User Manual

Allen-Bradley

SERCOS and Analog Motion Configuration and Startup

Catalog Numbers 1756-HYD02, 1756-M02AE, 1756-M02AS, 1756-M03SE, 1756-M08SE, 1756-M16SE, 1768-M04SE, 2094-SE02F-M00-S0, 2094-SE02F-M00-S1

Allen-Bradley • Rockwell Software

Rockwell Automation

Important user information

Read this document and the documents listed in the additional resources section about installation, configuration, and operation of this equipment before you install, configure, operate, or maintain this product. Users are required to familiarize themselves with installation and wiring instructions in addition to requirements of all applicable codes, laws, and standards.

Activities including installation, adjustments, putting into service, use, assembly, disassembly, and maintenance are required to be carried out by suitably trained personnel in accordance with applicable code of practice. If this equipment is used in a manner not specified by the manufacturer, the protection provided by the equipment may be impaired.

In no event will Rockwell Automation, Inc. be responsible or liable for indirect or consequential damages resulting from the use or application of this equipment.

The examples and diagrams in this manual are included solely for illustrative purposes. Because of the many variables and requirements associated with any particular installation, Rockwell Automation, Inc. cannot assume responsibility or liability for actual use based on the examples and diagrams.

No patent liability is assumed by Rockwell Automation, Inc. with respect to use of information, circuits, equipment, or software described in this manual.

Reproduction of the contents of this manual, in whole or in part, without written permission of Rockwell Automation, Inc., is prohibited.

Throughout this manual, when necessary, we use notes to make you aware of safety considerations.

	WARNING: Identifies information about practices or circumstances that can cause an explosion in a hazardous environment, which may lead to personal injury or death, property damage, or economic loss.
	ATTENTION: Identifies information about practices or circumstances that can lead to personal injury or death, property damage, or economic loss. Attentions help you identify a hazard, avoid a hazard, and recognize the consequence
Important:	Identifies information that is critical for successful application and understanding of the product.

Labels may also be on or inside the equipment to provide specific precautions.

	SHOCK HAZARD: Labels may be on or inside the equipment, for example, a drive or motor, to alert people that dangerous voltage may be present.
	BURN HAZARD: Labels may be on or inside the equipment, for example, a drive or motor, to alert people that surfaces may reach dangerous temperatures.
	ARC FLASH HAZARD: Labels may be on or inside the equipment, for example, a motor control center, to alert people to potential Arc Flash. Arc Flash will cause severe injury or death. Wear proper Personal Protective Equipment (PPE). Follow ALL Regulatory requirements for safe work practices and for Personal Protective Equipment (PPE).

Allen-Bradley, Rockwell Software, Rockwell Automation, and TechConnect are trademarks of Rockwell Automation, Inc.

Trademarks not belonging to Rockwell Automation are property of their respective companies.

This manual contains new and updated information. The following table contains the changes made to this revision.

Change	Topic
Updated the create a controller project information.	Create a controller project for Configure SERCOS Motion on page 21 and Create a controller project for Configure Analog Motion on page 43

Table of contents

Preface

Studio 5000 environment ... 13
What you need ... 13
Configuration and start-up scenarios ... 14
Description of the modules ... 14
Help for selecting drives and motors ... 15
Where to find sample projects ... 16
Additional resources ... 16
Legal Notices ... 18

Chapter 1

Configure SERCOS motion

Introduction for Configure SERCOS Motion ... 21
Create a controller project for Configure SERCOS Motion ... 21
Set time synchronization for Configure SERCOS Motion ... 24
Add a SERCOS motion module ... 25
Electronic Keying ... 26
Modify properties for a SERCOS motion module ... 27
More information ... 29
Add a SERCOS interface drive module ... 29
Modify properties for a SERCOS interface drive module ... 31
Add a motion group for Configure SERCOS Motion ... 33
Set the Base Update Period ... 34
Add an axis ... 36
Get axis information ... 37
Configure an axis for Configure SERCOS Motion ... 38
Set the homing sequence for Configure SERCOS Motion ... 41

Chapter 2

Configure analog motion

Introduction for Configure Analog Motion ... 43
Create a controller project for Configure Analog Motion ... 43
Set time synchronization for Configure Analog Motion ... 46
Add an analog module ... 47
Modify properties for an analog module ... 48
Add a hydraulic drive module ... 51
Modify properties for a hydraulic drive module ... 52
Configure the feedback type ... 54
Add a motion group for Configure Analog Motion ... 55
Set the Base Update Period ... 56
Add an axis for Configure Analog Motion ... 57

Get Axis Information ... 58
Configure an axis for Configure Analog Motion ... 59
Set the homing sequence for Configure Analog Motion ... 60

Chapter 3

Commission and tune

Introduction for Commission and Tune ... 61
Download a program to the controller ... 61
Test axis wiring and direction ... 61
Tune a SERCOS axis ... 63
Tune an analog axis ... 64
Troubleshoot faults ... 64
Manage motion faults ... 65
Configure the fault actions for an axis ... 66
Set the fault action for an axis ... 67
Inhibit an axis ... 67
When to inhibit an axis ... 67
Example 1 ... 68
Example 2 ... 68
Before you begin ... 68
Example: Inhibit an axis ... 70
Example: Uninhibit an axis ... 71
Test an axis with Motion Direct Commands ... 72
Access the Motion Direct Commands for a motion group ... 73
Access the Motion Direct Commands for an axis ... 73
Choose a command ... 74
Motion Direct Command dialog box ... 76
Motion Group Shutdown ... 77
Motion Direct Command error process ... 78
Motion Direct Command verification ... 78
Motion Direct Command execution error ... 80
What if the software goes offline or the controller changes modes? ... 80
Can two workstations give Motion Direct Commands? ... 80

Chapter 4

Program

Introduction ... 83
Definition of Jerk ... 83
Choose a profile ... 83
Use % of Time for the easiest programming of jerk ... 84
Velocity Profile Effects ... 85
Jerk Rate Calculation ... 86

Profile operand ... 90
Enter basic logic ... 94
Example: Motion control program ... 95
Download a program and run the logic ... 96
Choose a motion instruction ... 96
Sample projects ... 98
Troubleshoot axis motion ... 99
Why does my axis accelerate when I stop it? ... 99
Why does my axis overshoot its target speed? ... 102
Why is there a delay when I stop and then restart a jog? ... 105
Why does my axis overshoot its position and reverse direction? ... 107

Chapter 5

Home an axis

Introduction for Home an Axis ... 111
Guidelines for homing ... 111
Active homing ... 112
Passive homing ... 112
Absolute homing ... 113
Examples ... 113
Active homing examples ... 113
Passive homing examples ... 119
Homed Status ... 119
Feedback Integrity ... 120

Appendix A

Axis properties

Introduction for Axis Properties ... 121
General tab – AXIS_SERVO ... 121
General tab - AXIS_SERVO_DRIVE ... 122
Node with a Kinetix 6000 drive ... 123
General tab - AXIS_VIRTUAL ... 124
Motion Group ... 124
MOTION_GROUP structure ... 124
General tab – AXIS_GENERIC ... 126
Motion Planner tab ... 126
Units tab ... 128
Servo tab - AXIS_SERVO ... 128
Feedback Tab – AXIS_SERVO ... 129
Drive/Motor tab - AXIS_SERVO_DRIVE ... 132
Motor Feedback tab - AXIS_SERVO_DRIVE ... 137
Aux Feedback tab - AXIS_SERVO_DRIVE ... 137

Conversion tab ... 138
Homing tab - AXIS_SERVO ... 140
Homing tab - AXIS_SERVO_DRIVE ... 142
Homing tab - AXIS_VIRTUAL ... 144
Hookup tab - AXIS_SERVO ... 145
Hookup tab - AXIS_SERVO_DRIVE ... 146
Test Increment ... 147
Drive Polarity ... 147
Test Marker ... 147
Test Feedback ... 148
Test Command & Feedback ... 148
Tune tab - AXIS_SERVO, AXIS_SERVO_DRIVE ... 148
Travel Limit ... 148
Speed ... 149
Torque/Force (AXIS_SERVO_DRIVE) ... 149
Torque (AXIS_SERVO) ... 149
Direction ... 149
Damping Factor ... 150
Tune ... 150
Start Tuning ... 151
Dynamics tab - AXIS_SERVO, AXIS_SERVO _DRIVE, AXIS_VIRTUAL ... 152
Maximum Speed ... 152
Maximum Acceleration ... 153
Maximum Deceleration ... 153
Maximum Acceleration Jerk ... 153
Maximum Deceleration Jerk ... 154
Calculate button ... 154
Manual Adjust ... 155
Gains tab - AXIS_SERVO ... 156
Proportional (Position) Gain ... 156
Integral (Position) Gain ... 157
Differential ... 158
Proportional (Velocity) Gain ... 158
Integral (Velocity) Gain ... 158
Velocity Feedforward ... 159
Acceleration Feedforward ... 159
Integrator Hold ... 159
Manual Adjust ... 160
Gains Tab - AXIS_SERVO_DRIVE ... 160
Velocity Feedforward ... 161
Acceleration Feedforward ... 162

Proportional (Position) Gain 162
Integral (Position) Gain 162
Proportional (Velocity) Gain 163
Integral (Velocity) Gain 164
Integrator Hold 164
Manual Adjust 165
Set custom gains 165
Attribute 165
Output tab - AXIS_SERVO 166
Velocity Scaling 167
Torque/Force Scaling 167
Direction Scaling Ratio 168
Enable Low Pass Output Filter 168
Low-pass Output Filter Bandwidth 168
Manual Adjust 169
Output tab - AXIS_SERVO_DRIVE 169
Motor Inertia 170
Load Inertia Ratio 170
Torque/Force Scaling 170
Enable Notch Filter Frequency 171
Notch Filter Frequency 171
Enable Low Pass Output Filter 171
Low-pass Output Filter Bandwidth 171
Manual Adjust 172
Limits tab - AXIS_SERVO 172
Soft Travel Limits 173
Maximum Positive 173
Maximum Negative 173
Position Error Tolerance 174
Position Lock Tolerance 174
Output limit 174
Manual Adjust 175
Limits tab - AXIS_SERVO_DRIVE 175
Hard Travel Limits 176
Soft Travel Limits 176
Maximum Positive 176
Maximum Negative 176
Position Error Tolerance 177
Position Lock Tolerance 177
Peak Torque/Force Limit 177
Continuous Torque/Force Limit 177
Manual Adjust 178

Set custom limits....178
Attributes....179
Offset tab - AXIS_SERVO....180
Friction/Deadband Compensation and Backlash Compensation....181
Backlash Compensation Window....182
Backlash Compensation and Backlash Reversal Offset....182
Stabilization Window....183
Velocity Offset....183
Torque Offset....183
Output Offset....183
Manual Adjust....184
Offset tab - AXIS_SERVO_DRIVE....184
Backlash Compensation....185
Backlash Compensation Window....185
Backlash Compensation and Backlash Reversal Offset....186
Stabilization Window....186
Velocity Offset....186
Torque/Force Offset....187
Manual adjust for Offset tab....187
Fault Actions tab - AXIS_SERVO....187
Drive Fault....189
Feedback Noise....189
Feedback Loss....189
Position Error....189
Soft Overtravel....189
Fault Actions tab - AXIS_SERVO_DRIVE....190
Drive Enable Input....191
Drive Thermal....191
Motor Thermal....192
Feedback Noise....192
Feedback....192
Position Error....192
Hard Overtravel....192
Soft Overtravel....192
Phase Loss....193
Set custom stop action....193
Attributes....193
Tag tab....195
Name....195
Description....195
Tag Type....195
Data Type....195

Scope 196
Style 196
Monitoring axis tags 196
Create reports 196

Appendix B

Motion axis attributes

Introduction for Motion Axis Attributes 199
Accessing an MSG instruction 199
Interpreting the Attribute Tables 199
Replicated Attributes 200
Axis attributes 201
Additional error code information 293

Appendix C

Wiring diagrams

Introduction for Wiring Diagrams 295
1756-M02AE module 295
Notes 296
Ultra 100 Series Drive 296
Notes 296
Ultra 200 Series Drive 297
Notes 297
1398-CFLAExx cable 297
Pinouts for 1398-CFLAExx cable 298
Ultra3000 Drive 299
Ultra3000 to 1756-M02AE interconnect diagram 299
2090-U3AE-D44xx cable 300
1756-M02AS module 301
Wiring from AB 842A encoder without reset to 1756-M02AS RTB 302
Wiring for AB 842A encoder with remote reset to 1756-M02AS RTB 303
1756-HYD02 application example 304
1756-HYD02 module 305
Notes 305
LDTs 306
Temposonic GH feedback device 307
24V registration sensor 308
Notes 308
5V registration sensor 308
Notes 308

Home limit switch input 309
Notes 309
OK contacts 309
Notes 309

Appendix D

Servo loop block diagrams

Introduction for Servo Loop Block Diagrams 311
Interpreting the diagrams 311
AXIS_SERVO 312
Position servo with torque servo drive 312
Position servo with velocity servo drive 313
AXIS_SERVO_DRIVE 314
Motor Position Servo 314
Auxiliary Position Servo 315
Dual Position Servo 316
Motor Dual Command Servo 317
Auxiliary Dual Command Servo 318
Dual Command Feedback Servo 319
Velocity Servo 319
Torque Servo 319
Drive Gains 320

Index

This manual is a redesigned manual from publication LOGIX-UM002. A companion manual is available, which is Coordinate System User Manual, publication MOTION -UM002.

This manual is designed to give you the quickest and easiest approach to a SERCOS or Analog control solution. If you have any comments or suggestions, please see Documentation Feedback on the back cover of this manual.

Studio 5000 environment

The Studio 5000 Automation Engineering & Design Environment™ combines engineering and design elements into a common environment. The first element is the Studio 5000 Logix Designer™ application. The Logix Designer application is the rebranding of RSLogix™ 5000 software and will continue to be the product to program Logix5000™ controllers for discrete, process, batch, motion, safety, and drive-based solutions.

The Studio 5000® environment is the foundation for the future of Rockwell Automation® engineering design tools and capabilities. The Studio 5000 environment is the one place for design engineers to develop all elements of their control system.

What you need

You need the following to configure a SERCOS or Analog motion system.

SERCOS

- Logix L6*x* or Logix L7*x* controller
- SERCOS interface drive (6000,6200,2000, Ultra3000)
- SERCOS interface module
- Kinetix 6000 drive/actuators pair
- Logix Designer application

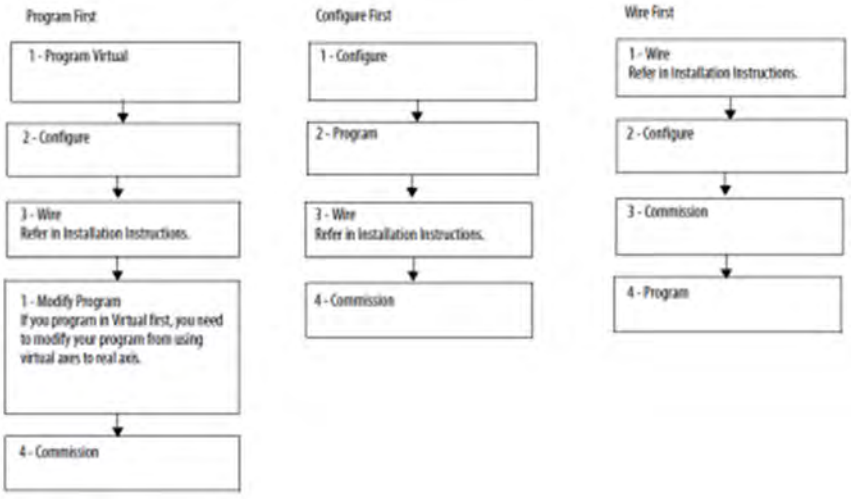
Analog

- Logix L6*x* controller
- Analog interface module
- Analog interface drive, Ultra3000
- Kinetix 6000 drive/actuators pair
- Logix Designer application

Configuration and start-up scenarios

The following are three example scenarios of how you can get a motion solution up and running.

Tip: Programming Virtual first is the safest method to begin with because you are separating the motion programming from the hardware.



Description of the modules

The following table describes the Logix5000 motion modules.

Motion Module	Description
1756-M03SE 1756-M08SE 1756-M16SE 1768-M04SE	Use a SERCOS interface module to connect the controller to SERCOS interface drives. • The SERCOS interface module lets you control digital drives by using high-speed, real time, serial communication. • SERCOS is the IEC 61491 Serial Real-time Communication System protocol over a fiber optic network. • The module uses a fiber optic network for all the wiring between the drives and the module.

Motion Module	Description
2094-SE02F-M00-S0, 2094-SE02F-M00-S1	Kinetix 6200 control modules use SERCOS interface to communicate with the Logix controller and EtherNet/IP to access the safety configuration tool.
1756-M02AE	The 1756-M02AE module is a two-axis servo module for drives/actuators that need a ±10V velocity or torque reference. Use the 1756-M02AE module when your equipment has quadrature encoder feedback. The module also has the following: • Home limit switch inputs • Drive fault inputs • Drive enable outputs • 5V or 24V position registration inputs • 250 µs position and velocity loop updates
1756-HYD02	The 1756-HYD02 module is a two-axis servo module for hydraulic actuators that need a ±10V velocity reference. Use the 1756-HYD02 module when your equipment has magnostrictive linear transducer (LDT) feedback. The module is similar to the 1756-M02AE module with the following exceptions. • Feed Forward adjust and single-step Auto Tune. • Gain ratio between extend direction and retract direction to accommodate hydraulic cylinder dynamics. • Intelligent transducer noise detection filtering in hardware and firmware replaces programmable IIR filtering.
1756-M02AS	The 1756-M02AS module is a two-axis servo module for drives/actuators that need a ±10V velocity or torque reference input. Use the 1756-M02AS module when your equipment has Serial Synchronous Input (SSI) position feedback. The module is similar to the 1756-M02AE module with these exceptions: • Gain ratio between extend direction and retract direction to accommodate hydraulic cylinder dynamics. • Intelligent transducer noise detection filtering in hardware and firmware replaces programmable IIR filtering. • SSI interface consisting of Differential Clock output and Data return signals replaces the differential encoder interface.

Help for selecting drives and motors

The Motion Analyzer utility helps you select the Rockwell Automation drives and motors based upon your load characteristics and typical motion application cycles.

You can access and download the program at the Motion Analyzer Software web page.

The Motion Analyzer guides you through wizard-like screens to collect information about your application. After you enter the information, for example, the load inertia, gear box ratio, feedback device, and brake requirements, the Motion Analyzer generates an easy-to-read list of recommended motors, drives, and other support equipment.

Where to find sample projects

To access the sample projects, take the following steps.

1. On the **Help** menu, click **Quick Start**.
2. On the **Quick Start** window, in the left navigation pane, expand **Controller Projects**, and click **Open Sample Project**.
3. In the **Open Project** dialog box, click **MSG_To_multiple_Controllers.acd**, and click **Open**.

Free sample code is also available at http://samplecode.rockwellautomation.com.

Additional resources

These documents contain additional information concerning related Rockwell Automation products. You can view or download publications at the Literature Library. To order paper copies of technical documentation, contact your local Rockwell Automation distributor or sales representative.

Resource	Description
Motion Coordinate System User Manual, publication MOTION-UM002.	Provides details on how to create and configure a coordinated motion system.
Logix5000 Controller Motion Instructions Reference Manual, publication MOTION-RM002.	Provides a programmer with details about motion instructions for a Logix-based controller.
Logix5000 Controllers Quick Start, publication 1756-QS001.	Describes how to get started programming and maintaining Logix5000 controllers.
Logix5000 Controllers Common Procedures, publication 1756-PM001.	Provides detailed and comprehensive information about how to program a Logix5000 controller.
Logix5000 Controllers General Instructions Reference Manual, publication 1756-RM003.	Provides a programmer with details about general instructions for a Logix-based controller.
Logix5000 Controllers Advanced Process Control and Drives Instructions Reference Manual, publication 1756-RM006.	Provides a programmer with details about process and drives instructions for a Logix-based controller.
PhaseManager User Manual, publication LOGIX-UM001.	Describes how to configure and program a Logix5000 controller to use equipment phases.
ControlLogix System User Manual, publication 1756-UM001.	Describes the necessary tasks to install, configure, program, and operate a ControlLogix system.

Resource	Description
CompactLogix Controllers User Manual, publication 1768-UM001.	Describes the necessary tasks to install, configure, program, and operate a CompactLogix system.
Analog Encoder (AE) Servo Module Installation Instructions, publication 1756-IN047.	Provides installation instructions for the Analog Encoder (AE) Servo Module, catalog number 1756-M02AE.
ControlLogix SERCOS interface Module Installation Instructions, publication 1756-IN572.	Provides installation instructions for the ControlLogix SERCOS interface modules, catalog number 1756-M03SE, 1756-M08SE, 1756-M16SE, 1756-M08SEG.
CompactLogix SERCOS interface Module Installation Instructions, publication 1768-IN005.	Provides installation instructions for the CompactLogix SERCOS interface Module, catalog number 1768-M04SE.
Ultra3000 Digital Servo Drives Installation Manual, publication 2098-IN003.	Provides the mounting, wiring, and connecting procedures for the Ultra3000 drives and standard Rockwell Automation/Allen-Bradley motors recommended for use with the Ultra3000 drives.
Ultra3000 Digital Servo Drives Integration Manual, publication 2098-IN005.	Provides powerup procedures, system integration, and troubleshooting tables for the Ultra3000 digital servo drives.
Kinetix 7000 High Power Servo Drive User Manual, publication 2099-UM001.	Provides details on how to plan for, mount, install, configure, and troubleshoot the Kinetix 7000 High Power Servo drive.
Kinetix 6000 Multi-axis Servo Drives User Manual, publication 2094-UM001.	Provides detailed installation instructions for mounting, wiring, and troubleshooting your Kinetix 6000 drive, and system integration for your drive/motor combination with a Logix controller.
Kinteix 6200 and Kinetix 6500 Safe Speed Monitoring Multi-axis Servo Drives Safety Reference Manual, publication 2094-RM001.	Provides information on wiring, configuring, and troubleshooting the safety functions of your Kinetix 6200 and Kinetix 6500 drives.
Kinetix 6200 and Kinetix 6500 Safe Torque-off Multi-axis Servo Drives Safety Reference Manual, publication 2094-RM002.	
8720MC High Performance Drives Installation manual, publication 8720MC-IN001.	Provides the mounting, wiring, and connecting procedures for the 8720MC and standard Rockwell Automation/Allen-Bradley motors recommended for use with the 8720MC drive.
8720MC High Performance Drives Integration manual, publication 8720MC-IN002.	Provides the startup, configuration, and troubleshooting procedures for the 8720MC drive.
Industrial Automation Wiring and Grounding Guidelines, publication 1770-4.1.	Provides general guidelines for installing a Rockwell Automation industrial system.
Product Certifications site.	Provides declarations of conformity, certificates, and other certification details.

Legal Notices

Copyright Notice

© 2015 Rockwell Automation, Inc. All rights reserved. Printed in USA.

This document and any accompanying Rockwell Software products are copyrighted by Rockwell Automation, Inc. Any reproduction and/or distribution without prior written consent from Rockwell Automation, Inc. is strictly prohibited. Please refer to the license agreement for details.

End User License Agreement (EULA)

You can view the Rockwell Automation End-User License Agreement ("EULA") by opening the License.rtf file located in your product's install folder on your hard drive.

Trademark Notices

Allen-Bradley, ControlBus, ControlFLASH, Compact GuardLogix, Compact I/O, ControlLogix, CompactLogix, DCM, DH+, Data Highway Plus, DriveLogix, DPI, DriveTools, Explorer, FactoryTalk, FactoryTalk Administration Console, FactoryTalk Alarms and Events, FactoryTalk Batch, FactoryTalk Directory, FactoryTalk Security, FactoryTalk Services Platform, FactoryTalk View, FactoryTalk View SE, FLEX Ex, FlexLogix, FLEX I/O, Guard I/O, High Performance Drive, Integrated Architecture, Kinetix, Logix5000, Logix5550, MicroLogix, DeviceNet, EtherNet/IP, PLC-2, PLC-3, PLC-5, PanelBuilder, PowerFlex, PhaseManager, POINT I/O, PowerFlex, Rockwell Automation, RSBizWare, Rockwell Software, RSEmulate, Historian, RSFieldbus, RSLinx, <RSKX>, RSNetWorx for DeviceNet, RSNetWorx for EtherNet/IP, RSMACC, RSView, RSView32, Rockwell Software Studio 5000 Automation Engineering & Design Environment, Studio 5000 View Designer, SCANport, SLC, SoftLogix, SMC Flex, Studio 5000, Ultra 100, Ultra 200, VersaView, WINtelligent, XM are trademarks of Rockwell Automation, Inc.

Any Rockwell Automation logo, software or hardware product not mentioned herein is also a trademark, registered or otherwise, of Rockwell Automation, Inc.

Other Trademarks

CmFAS Assistant, CmDongle, CmStick, CodeMeter, CodeMeter Control Center, and WIBU are trademarks of WIBU-SYSTEMS AG in the United States and/or other countries. Microsoft is a registered trademark of Microsoft Corporation in the United States and/or other countries. ControlNet is a trademark of ControlNet International. DeviceNet is a trademark of the Open DeviceNet Vendors Association (ODVA). Ethernet/IP is a trademark of ControlNet International under license by ODVA.

All other trademarks are the property of their respective holders and are hereby acknowledged.

Warranty

This product is warranted in accordance with the product license. The product's performance may be affected by system configuration, the application being performed, operator control, maintenance, and other related factors. Rockwell Automation is not responsible for these intervening factors. The instructions in this document do not cover all the details or variations in the equipment, procedure, or process described, nor do they provide directions for meeting every possible contingency during installation, operation, or maintenance. This product's implementation may vary among users.

This document is current as of the time of release of the product; however, the accompanying software may have changed since the release. Rockwell Automation, Inc. reserves the right to change any information contained in this document or the software at any time without prior notice. It is your responsibility to obtain the most current information available from Rockwell when installing or using this product.

Environmental Compliance

Rockwell Automation maintains current product environmental information on its website at http://www.rockwellautomation.com/rockwellautomation/about-us/sustainability-ethics/product-environmental-compliance.page

Contact Rockwell

Customer Support Telephone — 1.440.646.3434
Online Support — http://www.rockwellautomation.com/support/

Configure SERCOS motion

Introduction for Configure SERCOS Motion

Use this chapter for procedures on how to configure SERCOS motion control.

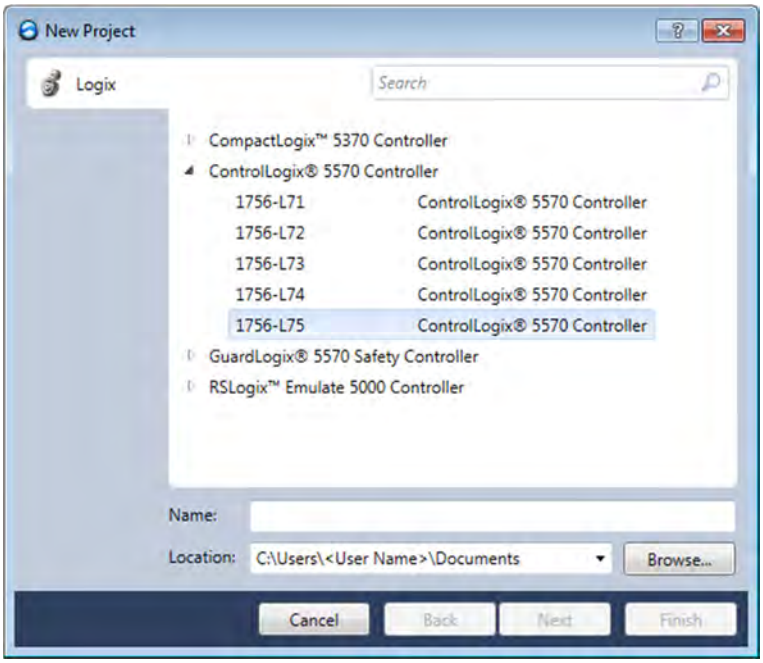
Create a controller project for Configure SERCOS Motion

Follow these instructions to create a controller project.

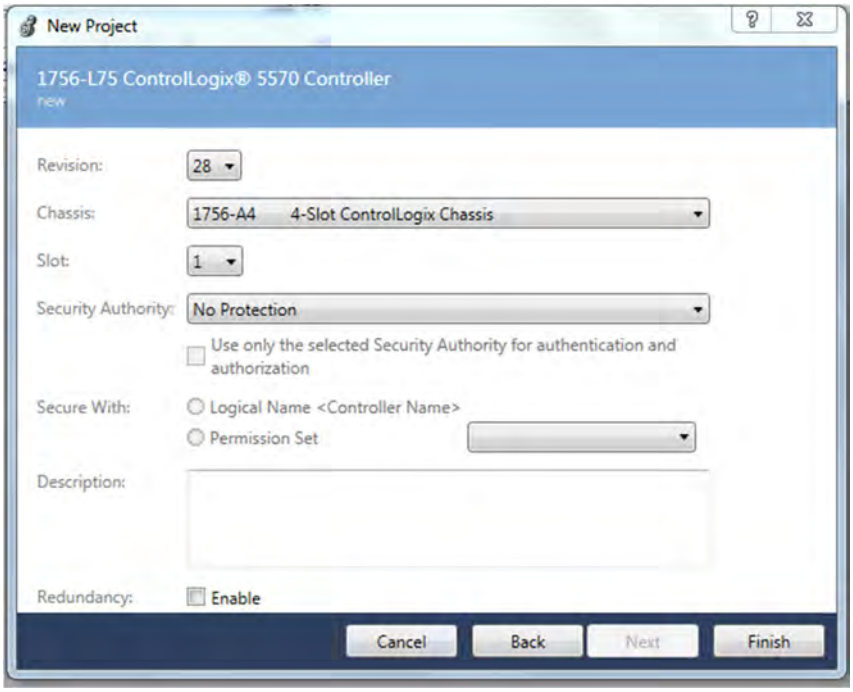
1. Open the Studio 5000 software.

2. In the Studio 5000 launcher, under **Create**, click **New Project**.

3. On the **New Project** dialog box, choose a controller.



4. In the **Name** box, type a name for the controller project, and then click **Next**.



5. In the **Revision** list, select the revision number for the controller.

6. In the **Chassis** list, select the type of chassis that holds the controller.

7. In the **Slot** list, select the physical slot where the controller is located.
8. In the **Security Authority** list, select a security option:
   - **No Protection** - All users can view and edit the project.
   - **FactoryTalk Security** - Only users authenticated through FactoryTalk Security can view and edit the project
9. (optional) Select **Use only the selected Security Authority for authentication and authorization** to associate this project with a specific Security Authority. When this check box is selected, users interacting with this project must be authenticated and authorized by the same Security Authority that was used to secure the project. Otherwise, unauthenticated users must rely on Guest User permissions.

   Tip: Guest User permissions are cached within the project. The Logix Designer application uses Guest User permissions when the project is opened but not connected to the FactoryTalk Security Authority that secures the project. By default, all Guest User permissions are denied. Guest User permissions are configured in the FactoryTalk Administration Console.

10. Select **Logical Name <Controller Name>** or **Permission Set** to apply specific permissions to the controller.

    Select **Logical Name <Controller Name>** to apply a Logical Name in FactoryTalk Services Platform that has the same name as the controller. If there is no existing Logical Name that matches the controller name, a new Logical Name is created with the controller's name. The new Logical Name inherits permissions from its parent resource. See FactoryTalk Help for more information on how networks and devices inherit security permissions.

    Select **Permission Set** to apply a specific set of permissions to the controller. The permission sets in the list are maintained in FactoryTalk Services Platform and identify a set of actions that are allowed or denied for a particular user and computer combination.

11. (optional) In the **Description** box, type a description for the controller.

    Tip: The description is limited to 128 bytes. Standard ASCII characters consume 1 byte per character, allowing for 128 characters. Characters in some languages require up to three bytes per character, resulting in less than 128 characters.

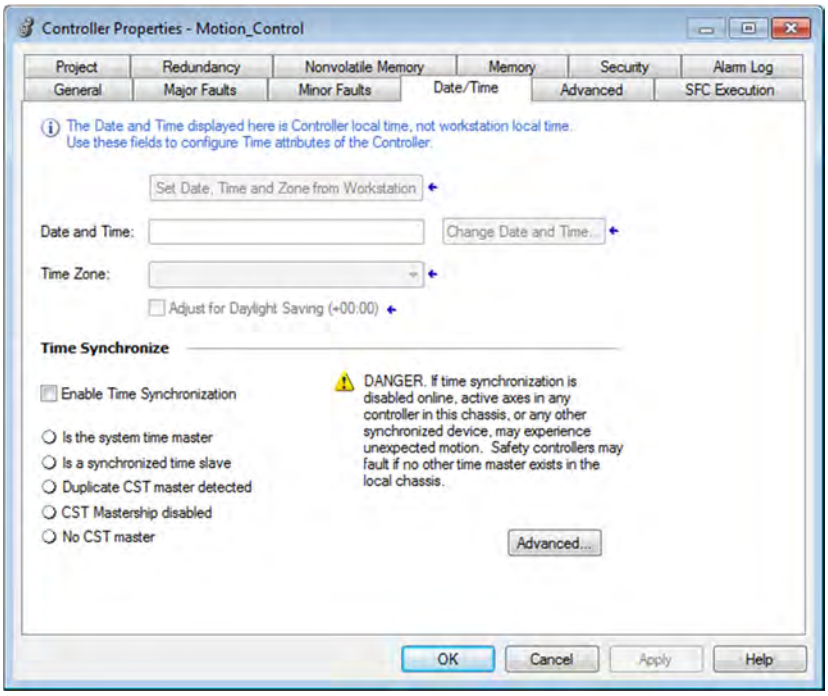
12. (optional) Select **Enable redundancy** if this project supports an automatic transfer of project control to a redundant controller in case of primary controller failure.
13. Click **Finish**.

Set time synchronization for Configure SERCOS Motion

Time Synchronization in ControlLogix is called CIP Sync. CIP Sync is a layer of functionality that Rockwell Automation has developed on top of the IEEE 1588 PTP protocol. CIP Sync lets you maintain accurate time synchronization of your automation solutions.

This setting establishes the module to participate in time synchronization. In systems with multiple processors, all controllers must have time synchronization enabled if they use CSmainT/PTP time. The 1756-EN*x*T communication modules win the arbitration over any processor.

1. In the **Controller Organizer**, double-click the controller.
2. On the **Controller Properties** dialog box, click the **Date/Time** tab.


Controller Properties - Motion_Control
Project
Redundancy
Nonvolatile Memory
Memory
Security
Alarm Log
General
Major Faults
Minor Faults
Date/Time
Advanced
SFC Execution
The Date and Time displayed here is Controller local time, not workstation local time.
Use these fields to configure Time attributes of the Controller.
Set Date, Time and Zone from Workstation
Date and Time:
Change Date and Time...
Time Zone:
Adjust for Daylight Saving (+00:00)
Time Synchronize
Enable Time Synchronization
Is the system time master
Is a synchronized time slave
Duplicate CST master detected
CST Mastership disabled
No CST master
DANGER. If time synchronization is disabled online, active axes in any controller in this chassis, or any other synchronized device, may experience unexpected motion. Safety controllers may fault if no other time master exists in the local chassis.
Advanced...
OK
Cancel
Apply
Help

3. Select the **Enable Time Synchronization** check box.
4. Click **OK**.

Without intervention, the Grandmaster is PTP and CST master. You can use the settings on the Advanced Dialog box to let this module win the arbitration over other processors and communication modules in the chassis.

See the Integrated Architecture and CIP Sync Configuration Application Technique, publication IA-AT003.

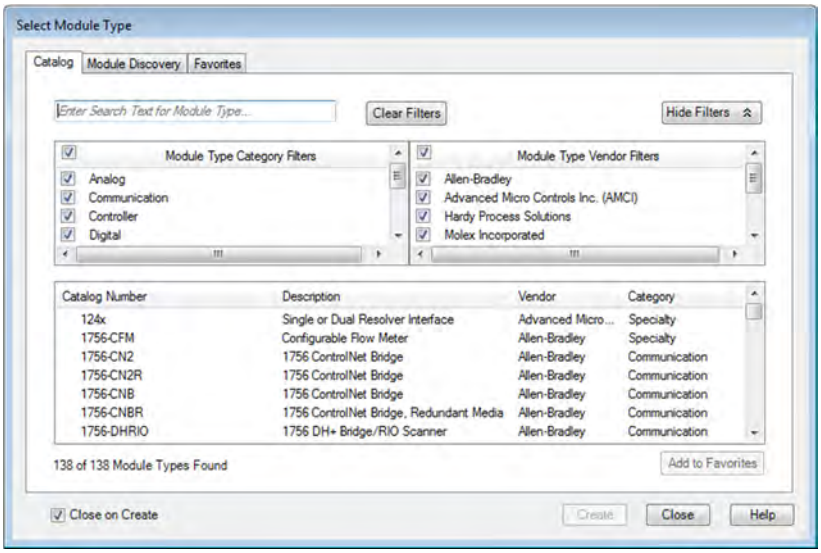
Add a SERCOS motion module

Follow these instructions to add a module to your system.

Important:	For all modules, use the firmware revision that goes with the firmware revision of your controller. See the release notes for your controller's firmware.

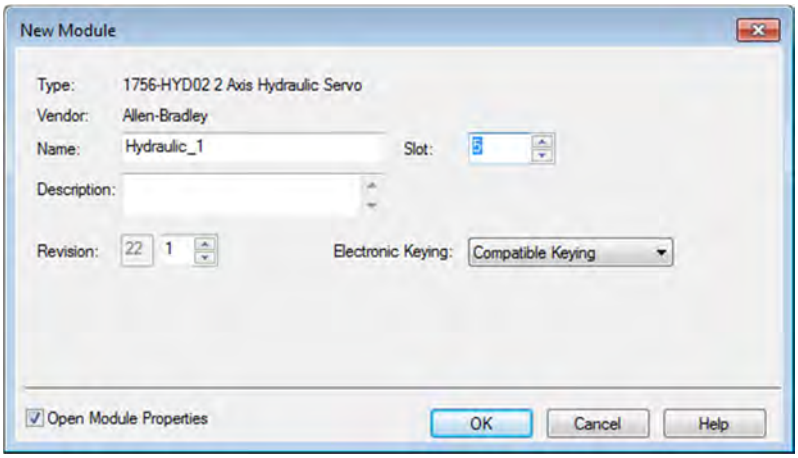
1. In the **Controller Organizer**, right-click the backplane and choose **New Module**.

2. On the **Select Module Type** dialog box, choose the module that you want to add.



3. Select the **Close on Create** check box, and click **Create**.

4. On the **New Module** dialog box, in the **Name** box, type a name for the module.



5. In the **Slot** box, choose the number that corresponds to the physical slot that contains the module.

6. (optional) In the **Description** box, type a description.

7. In the **Electronic Keying** list, choose a keying option of either **Compatible Keying** or **Exact Match** (see Electronic Keying on page 26).

WARNING: Disable Keying should **never** be used with motion modules.

8. Select the **Open Module Properties** check box, and click **OK**. Continue with the instructions to modify the properties for the motion module.

Electronic Keying

Electronic Keying reduces the possibility that you use the wrong device in a control system. It compares the device defined in your project to the installed device. If keying fails, a fault occurs. These attributes are compared.

Attribute	Description
Vendor	The device manufacturer.
Device Type	The general type of the device, for example, digital I/O module.
Product Code	The specific type of device. The Product Code maps to a catalog number.
Major Revision	A number that represents the functional capabilities of a device.
Minor Revision	A number that represents behavior changes in the device.

The following Electronic Keying options are available.

Keying Option	Description

Compatible Module	Lets the installed device accept the key of the device that is defined in the project when the installed device can emulate the defined device. With Compatible Module, you can typically replace a device with another device that has the following characteristics:
	• Same catalog number
	• Same or higher Major Revision
	• Minor Revision as follows:
	  • If the Major Revision is the same, the Minor Revision must be the same or higher.
	  • If the Major Revision is higher, the Minor Revision can be any number.

Disable Keying	Indicates that the keying attributes are not considered when attempting to communicate with a device. With Disable Keying, communication can occur with a device other than the type specified in the project. **ATTENTION:** Be extremely cautious when using Disable Keying; if used incorrectly, this option can lead to personal injury or death, property damage, or economic loss. We **strongly recommend** that you **do not use** Disable Keying. If you use Disable Keying, you must take full responsibility for understanding whether the device being used can fulfill the functional requirements of the application.
Exact Match	Indicates that all keying attributes must match to establish communication. If any attribute does not match precisely, communication with the device does not occur.

Carefully consider the implications of each keying option when selecting one.

Important:	Changing Electronic Keying parameters online interrupts connections to the device and any devices that are connected through the device. Connections from other controllers can also be broken. If an I/O connection to a device is interrupted, the result can be a loss of data.

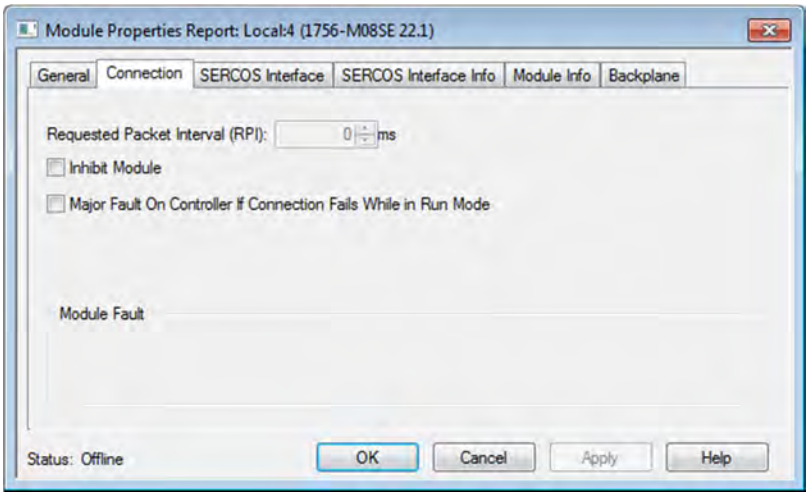
Modify properties for a SERCOS motion module

Follow these instructions to modify the module properties.

1. If the **Module Properties Report** dialog box is not already open, in the **Controller Organizer**, double-click the motion module.

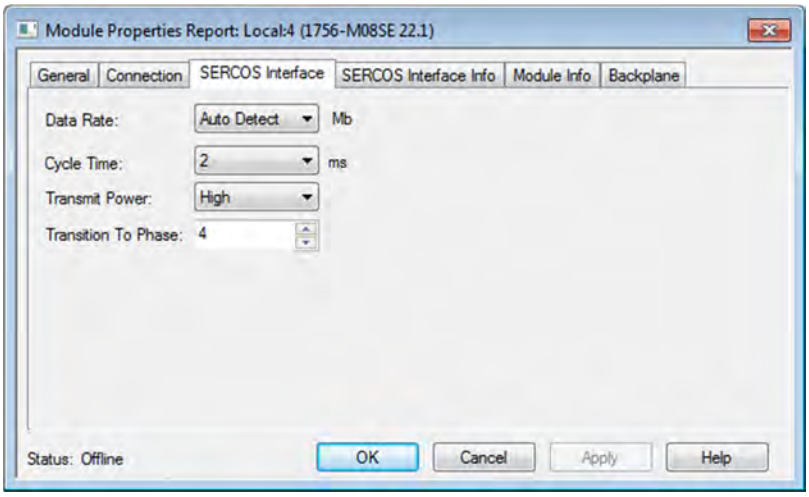
2. On the **Module Properties Report** dialog box, click the **Connection** tab.

3. Verify that the **Major Fault on Controller If Connection Fails While in Run Mode** check box is clear.



Tip: Keep this check box clear until you execute the program the first time. If this check box is selected, then a Major Fault is generated when the SERCOS ring attempts to phase up the first time.

4. Click the **SERCOS Interface** tab.



5. In the **Data Rate** and **Cycle Time** boxes, choose the baud rate and update rate for the SERCOS ring using the following table as a guide. The data rate and the number of drives associated with the module dictate your minimum cycle time.

Baud Rate of Drives	Number of Drives on the Ring	Type of Drives	Cycle Time
4 Mbps	Up to 2	Kinetix 2000 Kinetix 6000 Kinetix 6200 Kinetix 7000	0.5 ms

Baud Rate of Drives	Number of Drives on the Ring	Type of Drives	Cycle Time
	Up to 4	—	1 ms
	Up to 8	—	2 ms
	9…16	—	No Support
Tip:	The Ultra 3000 must be 1 or 2 ms SERCOS cycle time because the position loop update time minimum is 1 ms.		
8 Mbps	Up to 4	Kinetix 2000 Kinetix 6000 Kinetix 6200 Kinetix 7000	0.5 ms
	Up to 8	—	1 ms
	Up to 16	—	2 ms

For more information about configuring the Kinetix 6200 and Kinetix 6500 servo drive systems, see the Kinetix 6200 and Kinetix 6500 Modular Multi-axis Servo Drive User Manual, publication 2094-UM002.

6. Click **OK**.

More information

For more detailed information on Electronic Keying, see Electronic Keying in Logix5000 Control Systems Application Technique, publication LOGIX-AT001.

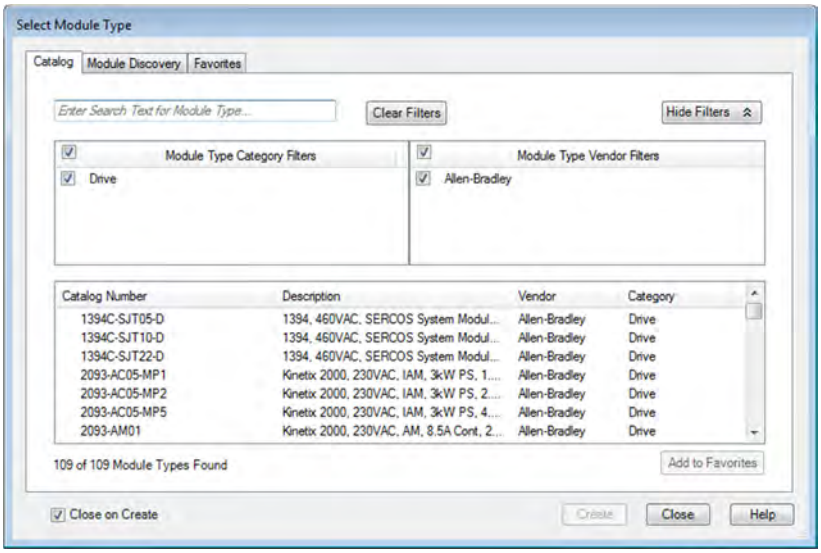
Add a SERCOS interface drive module

Follow these instructions to add a SERCOS interface drive module to the I/O configuration of the controller.

Important:	For all modules, use the firmware revision that goes with the firmware revision of your controller. See the release notes for your controller's firmware.

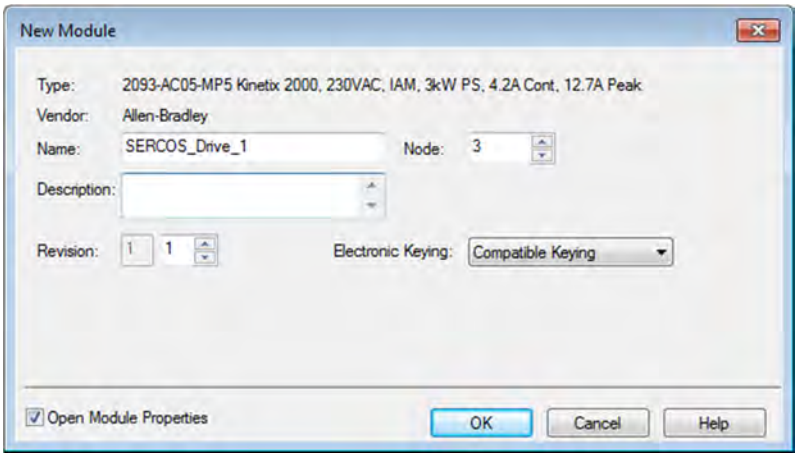
1. In the **Controller Organizer**, right-click **SERCOS Network** and choose **New Module**.

2. On the **Select Module Type** dialog box, choose the module that you want to add.



3. Select the **Close on Create** check box, and click **Create**.

4. On the **New Module** dialog box, in the **Name** box, type a name for the interface drive.



5. In the **Node** box, choose the SERCOS node number of the interface drive (see the module's rotary switch).

6. (optional) In the **Description** box, type a description.

7. In the **Revision** boxes, select the Major Revision and Minor Revision of the module.

8. In the **Electronic Keying** list, choose a keying option of either **Compatible Keying** or **Exact Match**. See Electronic Keying on page 26.

WARNING: Disable Keying should **never** be used with motion modules.

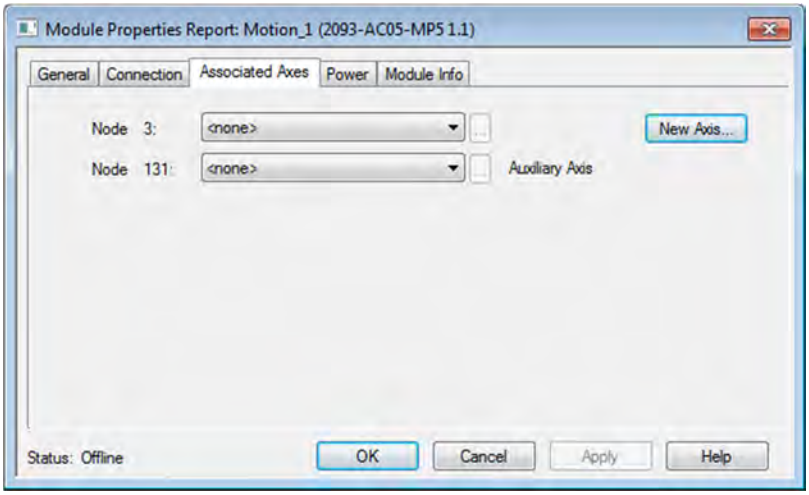
9. Select the **Open Module Properties** check box, and click **OK**. Continue with the instructions to modify the properties for the interface drive.

Modify properties for a SERCOS interface drive module

Follow these instructions to modify drive properties.

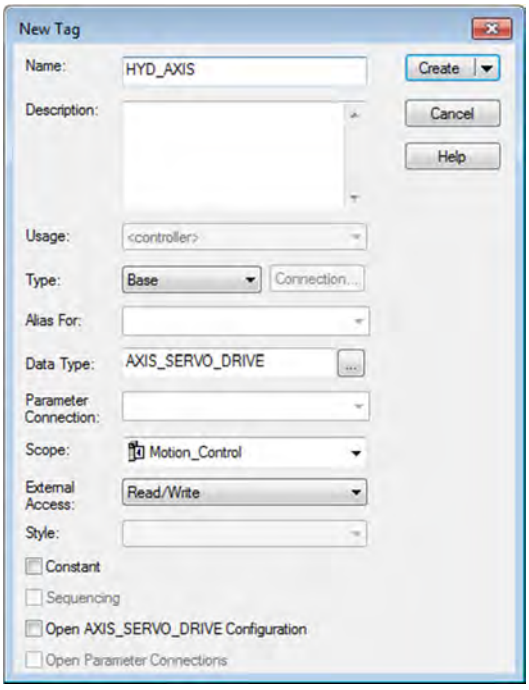
1. If the **Module Properties Report** dialog box is not already open, in the **Controller Organizer**, double-click the interface drive.

2. On the **Module Properties Report** dialog box, click the **Associated Axes** tab.



3. Click **New Axis** to create an AXIS_SERVO tag to associate to one of the nodes.

4. On the **New Tag** dialog box, in the **Name** box, type a name for the axis tag.



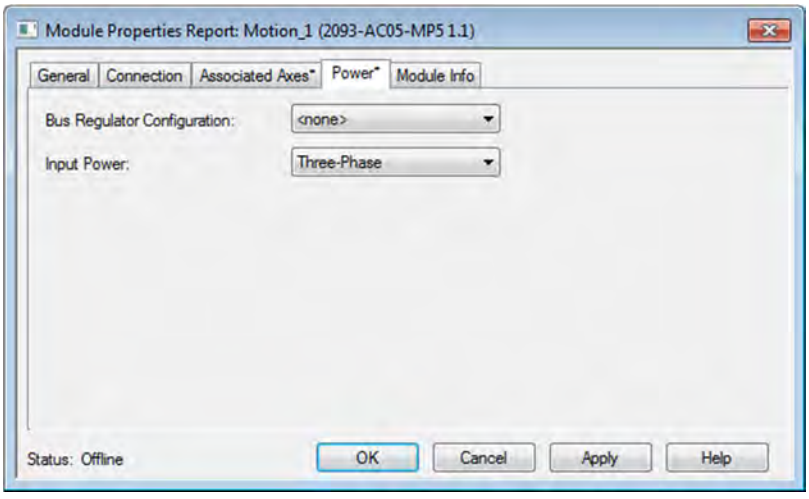
5. Click **Create**.

6. (optional) Repeat steps 3 through 5 if an additional axis is required.

7. On the **Module Properties Report** dialog box, in the **Node** (Number) box, choose an AXIS_SERVO_DRIVE tag to associate with the drive's node.

8. (optional) In the second **Node** (Number) box, choose an axis.

9. Click the **Power** tab.



10. In the **Bus Regulator Configuration** box, choose the catalog number that describes the bus regulator device used by the drive module. Depending upon the drive you have selected, one or more of the bus regulator IDs may be available.

Important:	This setting does not apply to the 8720MC drive, the Ultra3000 SERCOS drive, or Kinetix 6200 Integrated Drive Motor (IDM) drive. For those drives, <none> is the only available option.

11. If you are configuring a Kinetix 2000 230-volt drive, in the **Input Power** box, choose **Single-Phase** or **Three-Phase**.

12. If you are configuring an IDM Power Interface Module (IPIM) or IDM module, in the **Additional Bus Capacitance** box, choose the additional bus capacitance that is required. This parameter is only valid if an axis is associated with the Kinetix 6000 and Kinetix 6200 IAM module. When an axis is initially associated to the IAM, this parameter shows the default value and is enabled. If a valid value is set in the associated axis, and the axis is not associated to the IAM, this parameter is reset to the default value and cannot be changed until an axis is associated to the IAM.

13. Click **OK**.

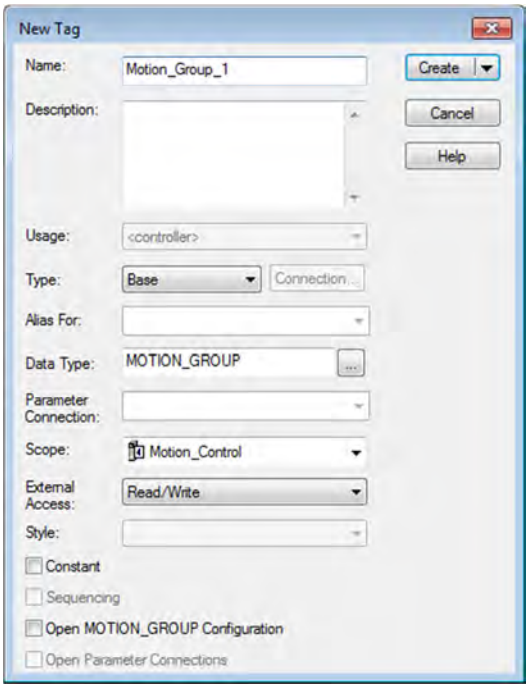
Add a motion group for Configure SERCOS Motion

Follow these instructions to add a motion group.

Important:	Only one motion group can be created for each project.

1. In the **Controller Organizer**, right-click **Motion Groups** and choose **New Motion Group**.

2. On the **New Tag** dialog box, in the **Name** box, type a name for the motion group.



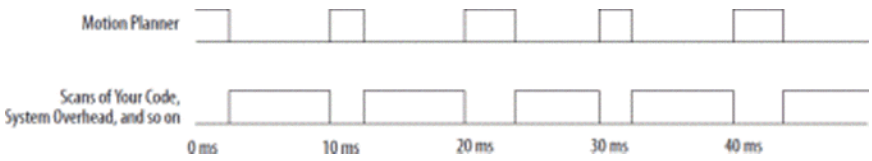
3. (optional) In the **Description** box, type a description.

4. Click **Create**.

Set the Base Update Period

The Base Update Period (also known as the Coarse Update Period) is how often the motion planner runs. The motion planner is the part of the controller that handles position and velocity information for the axes. When the motion planner runs, it interrupts most other tasks regardless of their priority.

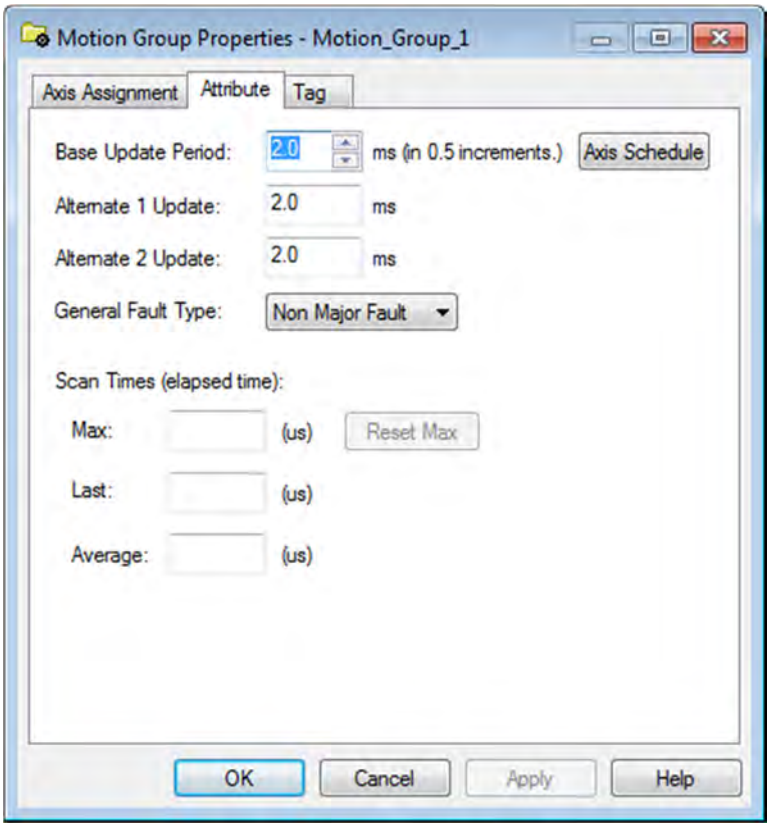
Example: If the Base Update Period is set to 10 ms, then every 10 ms the controller stops scanning the code and performing other system overhead tasks, and runs the motion planner.



Use the following guidelines to set the Base Update Period.

Guideline	Description
Number of Axes	1756-L6*x* controller 4 axes/ms 1756-L7*x* controller8 axes/ms
Save Controller's Time	Leave at least half the controller's time for the scan of all your code.
Base Update Period and SERCOS modules	If you have SERCOS interface motion modules, set the Base Update Period to a multiple of the cycle time of the motion module. Example: If the cycle time is 2 ms, set the Base Update Period to 8 ms, 10 ms, 12ms, and so on.
Base Update Period and Analog modules	If you have analog motion modules, set the Base Update Period to: • at least 3 times the servo update period of the motion module. • a multiple of the servo update period of the motion module.

1. In the **Controller Organizer**, double-click the motion group.
2. Click the **Attribute** tab.



3. In the **Base Update Period** box, choose the update period using the guidelines mentioned earlier. The valid values range from **0.5** to **32**, in 0.5 increments.

   Tip: The **Axis Schedule** button opens the **Axis Schedule** dialog box where the base and alternate update periods can be scheduled and assigned to axes. Since axes used in coordinate system objects cannot be multiplexed, only the Base Update Period is used. Therefore, there is no need to open the **Axis Schedule** dialog box.

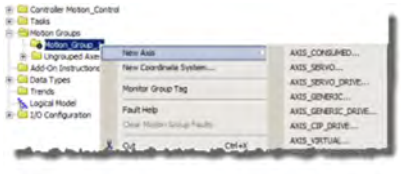
4. In the **General Fault Type** list, choose **Non Major Fault**.

5. Click **OK**.

Add an axis

Follow these instructions to add an axis for each of your drives.

1. In the **Controller Organizer**, right-click the motion group and choose **New Axis**.

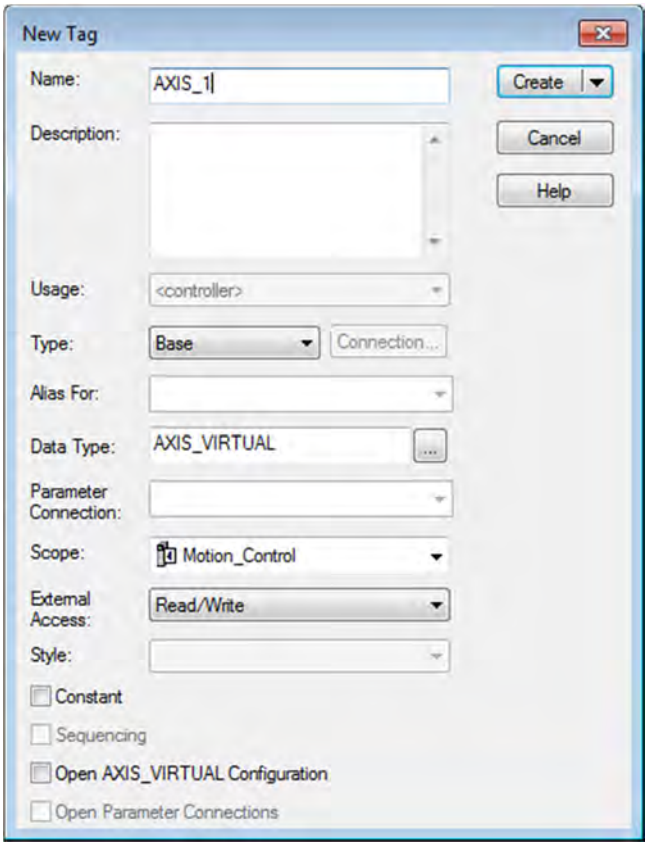


2. Choose the data type based on the following guidelines.

   - If you use one of the following motion modules, choose **AXIS_SERVO_DRIVE**.
     - 1756-M03SE
     - 1756-M08SE
     - 1756-M16SE
     - 1768-M04SE

- If you want to use a virtual configuration (no hardware), choose **AXIS_VIRTUAL**.

3. On the **New Tag** dialog box, in the **Name** box, type a name for the axis.



4. (optional) In the **Description** box, type a description for the axis.

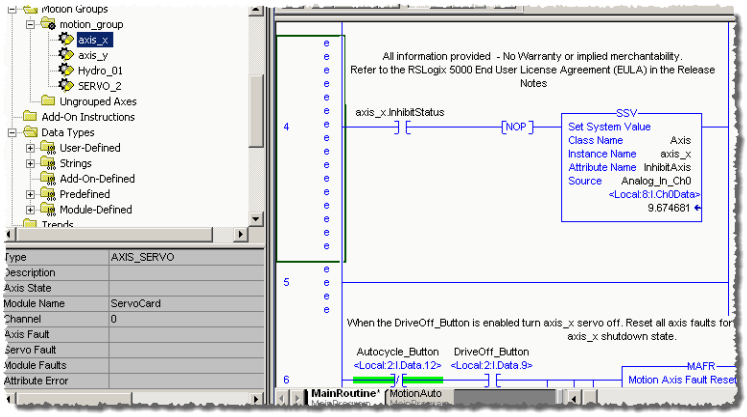
5. Click **Create**.

Get axis information

You can get information about an axis in several ways.

- Use the Quick View pane to see the state and faults of an axis.
- Use a Get System Value (GSV) instruction or Set System Value (SSV) instruction to read or change the configuration at run-time.

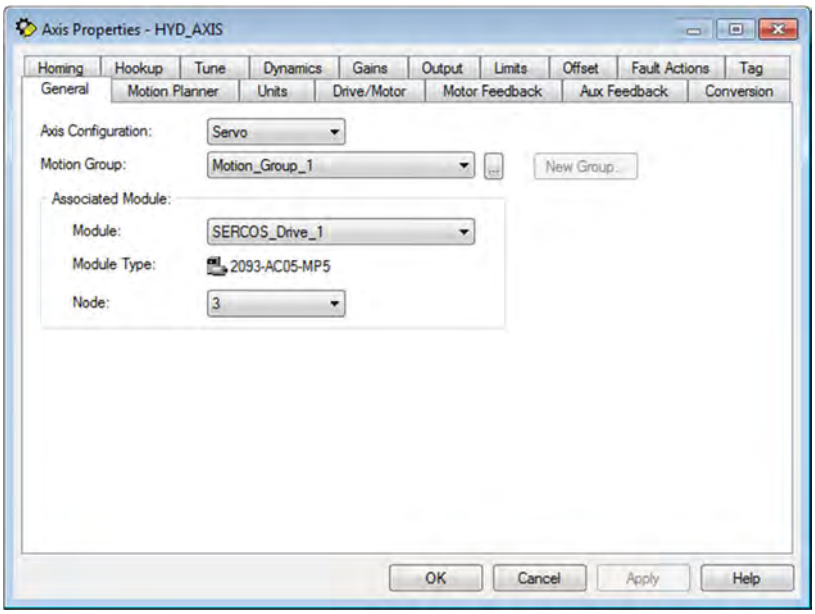
- Use the tag of the axis for status and faults.



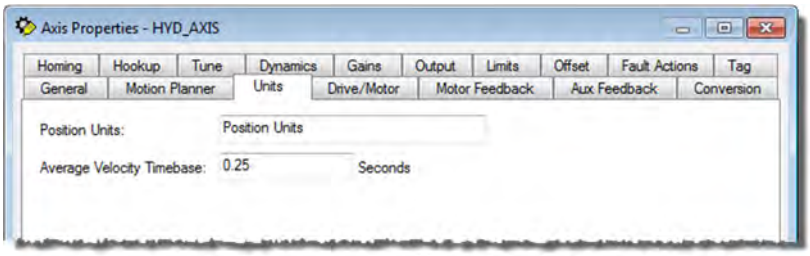
Configure an axis for Configure SERCOS Motion

Follow these instructions to configure the axis of a SERCOS interface drive. The steps are may differ depending on the type of drive you are configuring.

1. In the **Controller Organizer**, double-click the axis.
2. On the **General** tab, verify that the assigned motion group and module information is correct. Make any needed changes.

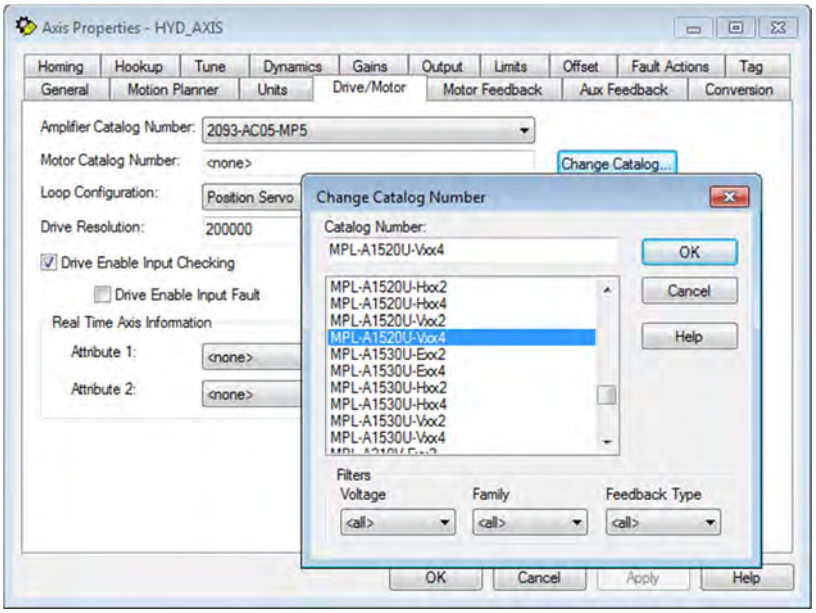


3. Click the **Units** tab, and in the **Position Units** and **Average Velocity Timebase** boxes, enter the values.



4. Click the **Drive/Motor** tab.

5. Click **Change Catalog**, and on the **Change Catalog Number** dialog box, choose the related motor catalog numbers, and click **OK**.



The following conditions apply for the Integrated Drive Motor (IDM).

- The **Amplifier Catalog Number** appears dimmed and cannot be edited when the axis is associated to an IDM.
- The **Motor Catalog Number** is automatically set to the catalog number of the only motor available from the Motion Database for the IDM module when it is associated to the axis.
- The Catalog Number for the IDM appears dimmed and is read-only. It is automatically set to the compatible motor from the motion database for each IDM.
- An IDM does not support Auxiliary Axis loop configurations. When the **Axis Configuration** box on the **General** tab is set to **Servo**, the selections for **Loop Configuration** for the IDM are:
  - **Position Servo**
  - **Dual Command Servo (does not use the Auxiliary Port)**
  - **Velocity Servo**
  - **Torque Servo**

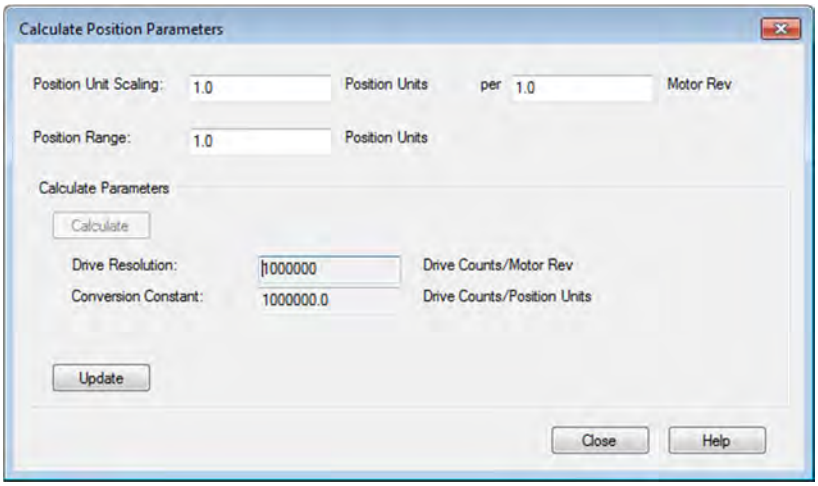
  When the **Axis Configuration** box on the **General** tab is set to **Feedback Only**, the selection for **Loop Configuration** for the IDM is **Motor Feedback Only**.
- The following are real-time attributes for the IDM.
  - Position Command
  - Position Feedback
  - Position Error
  - Position Int. Error
  - Velocity Command
  - Velocity Feedback
  - Velocity Error
  - Velocity Int. Error
  - Accel Command
  - Accel Feedback
  - Marker Distance
  - Torque Command
  - Torque Feedback
  - Positive Dynamic Torque Limit
  - Negative Dynamic Torque Limit
  - Motor Capacity
  - Drive Capacity
  - Power Capacity

- Bus Regulator Capacity
- Motor Electrical Angle
- Torque Limit Source
- DC Bus Voltage

6. In the **Drive Resolution** box, type the resolution, and then in the **Drive Counts /** box, select the units.

7. Click **Calculate**.

8. On the **Calculate Position Parameters** dialog box, click **Calculate**, review the values and adjust as needed until you have the desired calculated values, then click **Update** and **Close**.

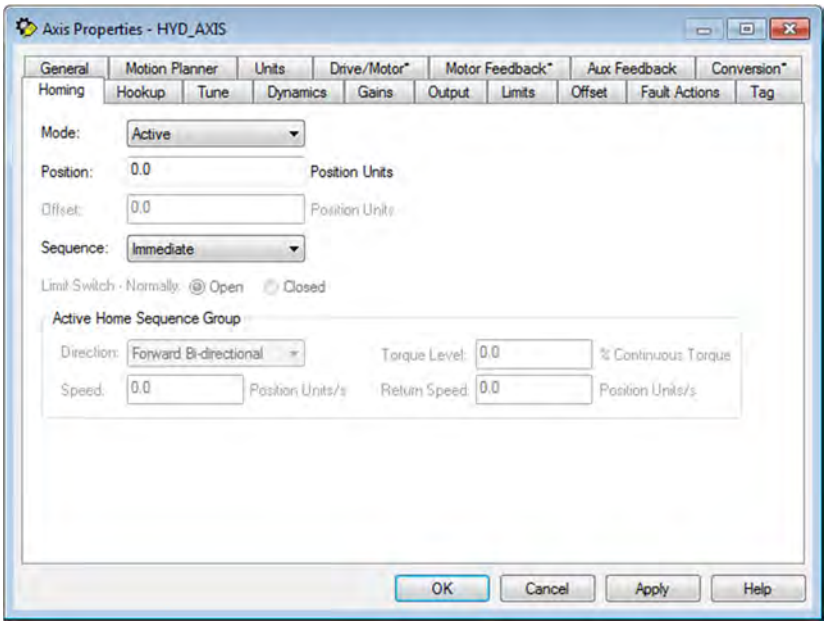


Set the homing sequence for Configure SERCOS Motion

For complete information about Homing modes, methods, and guidelines, see Home an Axis on page 111.

Follow these instructions to set the homing sequence.

1. Click the **Homing** tab, and in the **Mode** box, choose the homing mode.



2. In the **Position** box, type the position units.

3. In the **Sequence** box, choose the sequence type.

4. For all sequence types except **Intermediate**, in the **Description** box, choose the active home sequence type, and then in the **Speed** and **Return Speed** boxes, set the homing speeds.

5. Click **OK** to apply the changes.

Configure analog motion

Introduction for Configure Analog Motion

Use this chapter for step-by-step procedures on how to configure analog motion control.

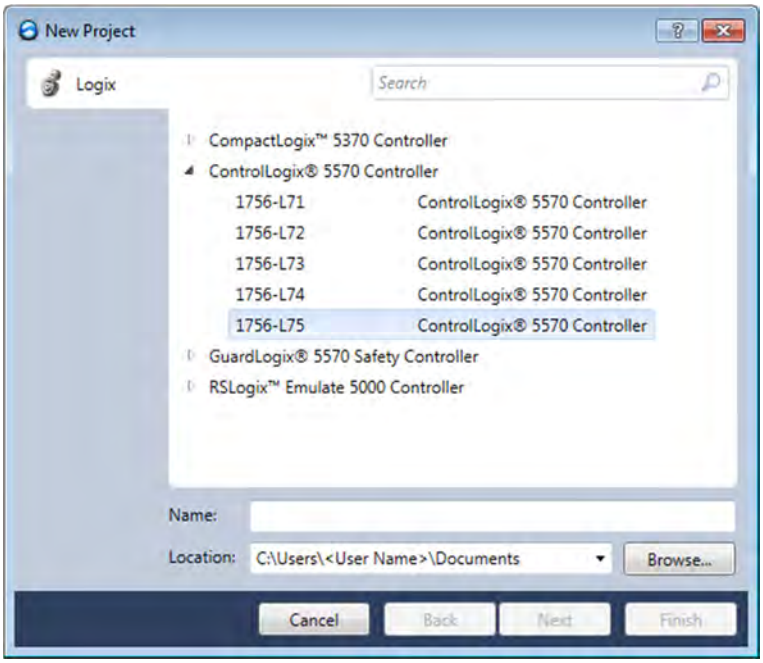
Create a controller project for Configure Analog Motion

Follow these instructions to create a controller project.

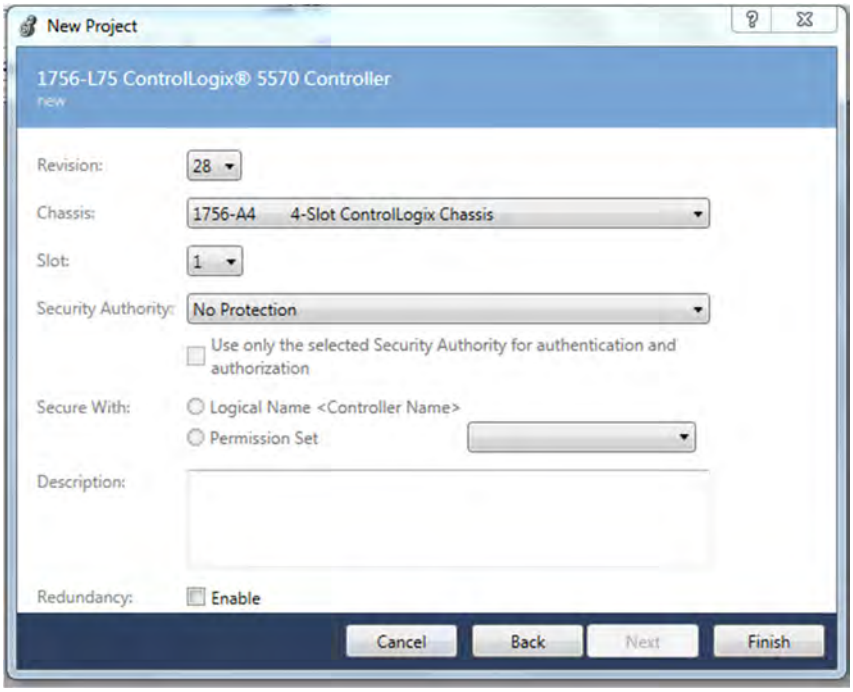
1. Open the Studio 5000 software.

2. In the Studio 5000 launcher, under **Create**, click **New Project**.

3. On the **New Project** dialog box, choose a controller.



4. In the **Name** box, type a name for the controller project, and then click **Next**.



5. In the **Revision** list, select the revision number for the controller.

6. In the **Chassis** list, select the type of chassis that holds the controller.

7. In the **Slot** list, select the physical slot where the controller is located.
8. In the **Security Authority** list, select a security option:
   - **No Protection** - All users can view and edit the project.
   - **FactoryTalk Security** - Only users authenticated through FactoryTalk Security can view and edit the project
9. (optional) Select **Use only the selected Security Authority for authentication and authorization** to associate this project with a specific Security Authority. When this check box is selected, users interacting with this project must be authenticated and authorized by the same Security Authority that was used to secure the project. Otherwise, unauthenticated users must rely on Guest User permissions.

   Tip: Guest User permissions are cached within the project. The Logix Designer application uses Guest User permissions when the project is opened but not connected to the FactoryTalk Security Authority that secures the project. By default, all Guest User permissions are denied. Guest User permissions are configured in the FactoryTalk Administration Console.

10. Select **Logical Name <Controller Name>** or **Permission Set** to apply specific permissions to the controller.

    Select **Logical Name <Controller Name>** to apply a Logical Name in FactoryTalk Services Platform that has the same name as the controller. If there is no existing Logical Name that matches the controller name, a new Logical Name is created with the controller's name. The new Logical Name inherits permissions from its parent resource. See FactoryTalk Help for more information on how networks and devices inherit security permissions.

    Select **Permission Set** to apply a specific set of permissions to the controller. The permission sets in the list are maintained in FactoryTalk Services Platform and identify a set of actions that are allowed or denied for a particular user and computer combination.

11. (optional) In the **Description** box, type a description for the controller.

    Tip: The description is limited to 128 bytes. Standard ASCII characters consume 1 byte per character, allowing for 128 characters. Characters in some languages require up to three bytes per character, resulting in less than 128 characters.

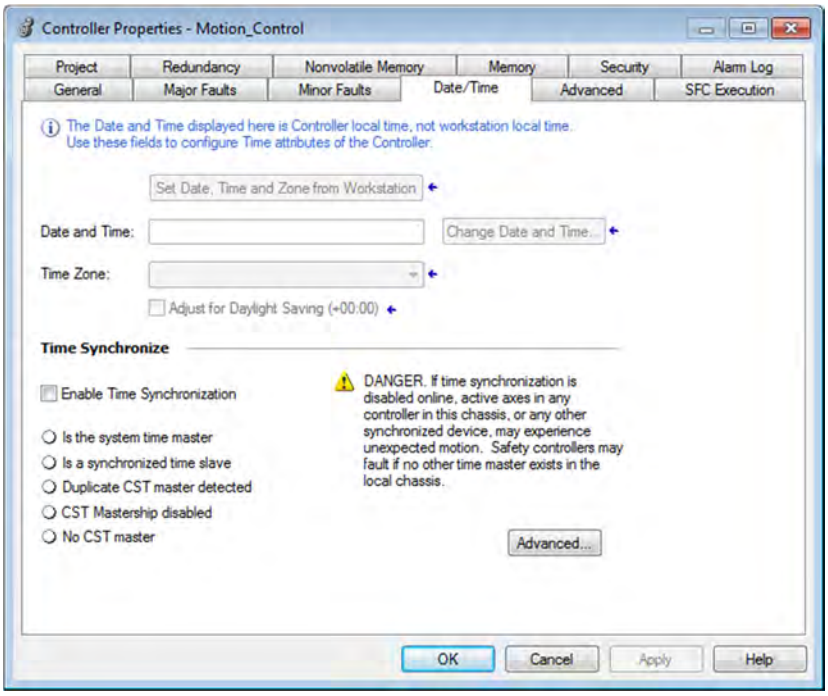
12. (optional) Select **Enable redundancy** if this project supports an automatic transfer of project control to a redundant controller in case of primary controller failure.
13. Click **Finish**.

Set time synchronization for Configure Analog Motion

Time Synchronization in ControlLogix is called CIP Sync. CIP Sync is a layer of functionality that Rockwell Automation has developed on top of the IEEE 1588 PTP protocol. CIP Sync lets you maintain accurate time synchronization of your automation solutions.

This setting establishes the module to participate in time synchronization. In systems with multiple processors, all controllers must have time synchronization enabled if they use CSmainT/PTP time. The 1756-EN*x*T communication modules win the arbitration over any processor.

1. In the **Controller Organizer**, double-click the controller.
2. On the **Controller Properties** dialog box, click the **Date/Time** tab.


Controller Properties - Motion_Control
Project
Redundancy
Nonvolatile Memory
Memory
Security
Alarm Log
General
Major Faults
Minor Faults
Date/Time
Advanced
SFC Execution
The Date and Time displayed here is Controller local time, not workstation local time.
Use these fields to configure Time attributes of the Controller.
Set Date, Time and Zone from Workstation
Date and Time:
Change Date and Time...
Time Zone:
Adjust for Daylight Saving (+00:00)
Time Synchronize
Enable Time Synchronization
Is the system time master
Is a synchronized time slave
Duplicate CST master detected
CST Mastership disabled
No CST master
DANGER. If time synchronization is disabled online, active axes in any controller in this chassis, or any other synchronized device, may experience unexpected motion. Safety controllers may fault if no other time master exists in the local chassis.
Advanced...
OK
Cancel
Apply
Help

3. Select the **Enable Time Synchronization** check box.
4. Click **OK**.

Without intervention, the Grandmaster is PTP and CST master. You can use the settings on the Advanced Dialog box to let this module win the arbitration over other processors and communication modules in the chassis.

See the Integrated Architecture and CIP Sync Configuration Application Technique, publication IA-AT003.

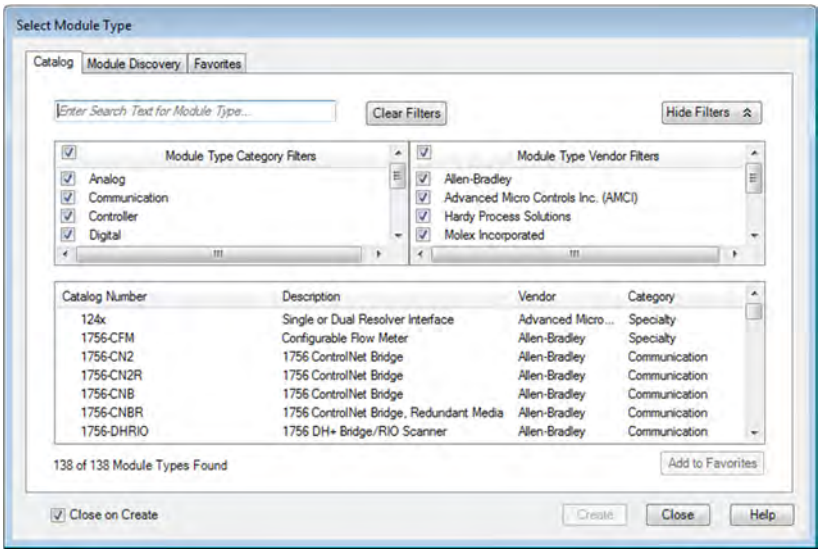
Add an analog module

Follow these instructions to add an analog module to your system.

Important:	For all modules, use the firmware revision that goes with the firmware revision of your controller. See the release notes for your controller's firmware.

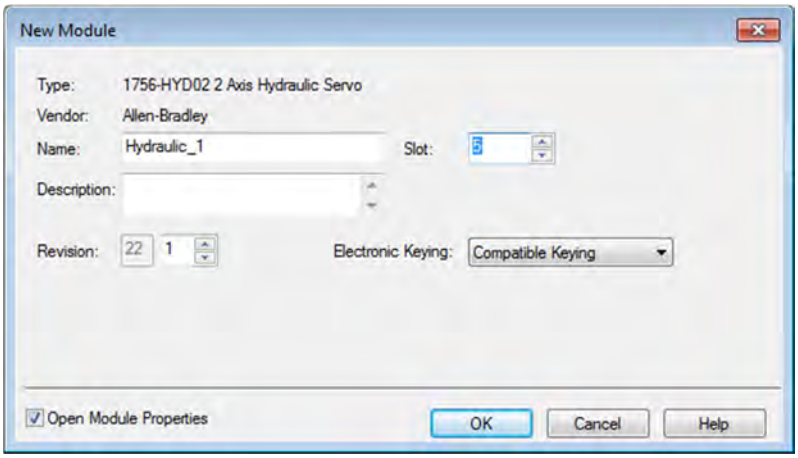
1. In the **Controller Organizer**, right-click the backplane and choose **New Module**.

2. On the **Select Module Type** dialog box, choose the module that you want to add.



3. Select the **Close on Create** check box, and click **Create**.

4. On the **New Module** dialog box, in the **Name** box, type a name for the module.



5. In the **Slot** box, choose the number that corresponds to the physical slot that contains the module.

6. (optional) In the **Description** box, type a description.

7. In the **Electronic Keying** box, choose a keying option of either **Compatible Keying** or **Exact Match** (see Electronic Keying on page 26).

WARNING: Disable Keying should **never** be used with motion modules.

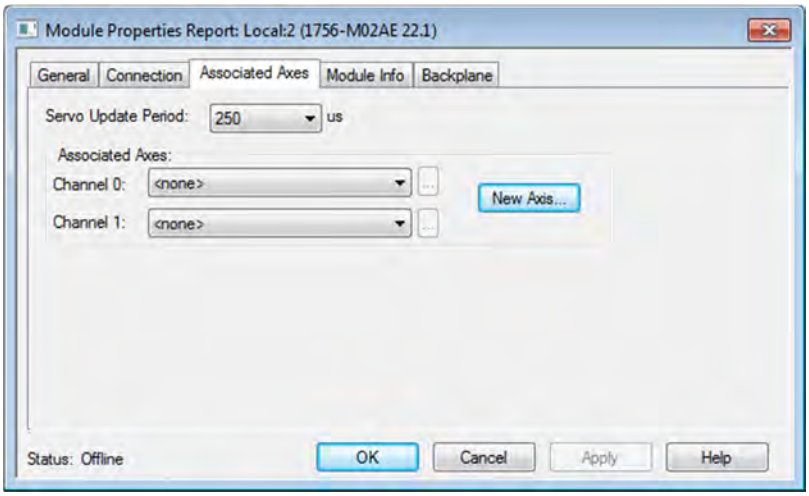
8. Select the **Open Module Properties** check box, and click **OK**. Continue with the procedure to modify the properties for the module.

Modify properties for an analog module

Follow these instructions to modify the properties for an analog module.

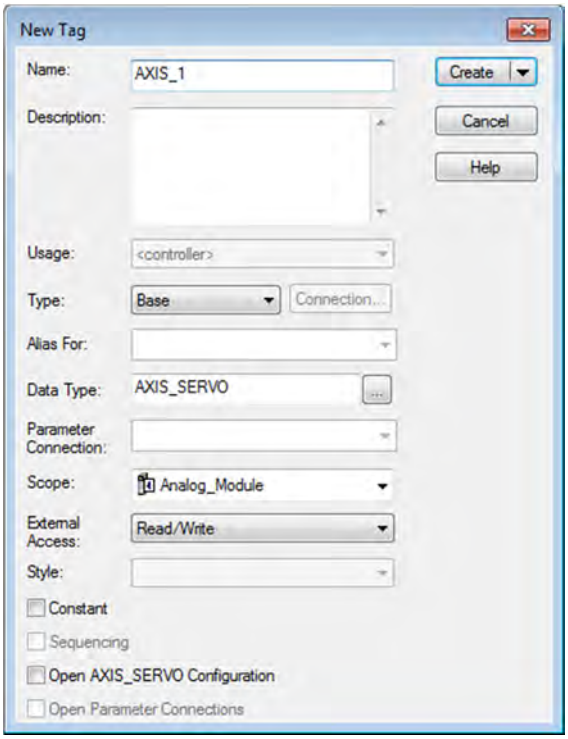
1. In the **Controller Organizer**, double-click the module.

2. On the **Module Properties Report** dialog box, click the **Associated Axes** tab.



3. Click **New Axis** to create an axis to associate with this module.

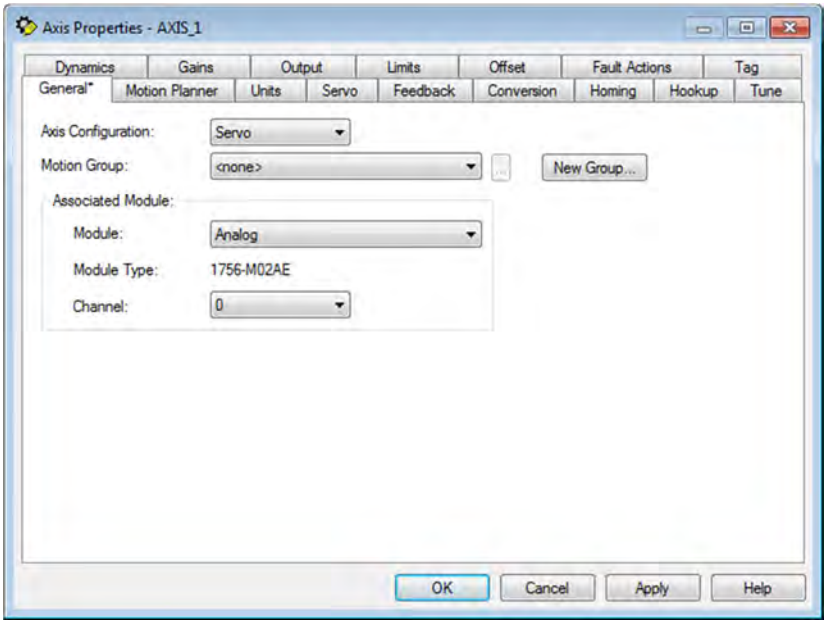
4. On the **New Tag** dialog box, type a name for the Axis and click **Create.**



5. On the **Module Properties Report** dialog box, in the **Channel 0** box, choose the new axis to assign it to the module.

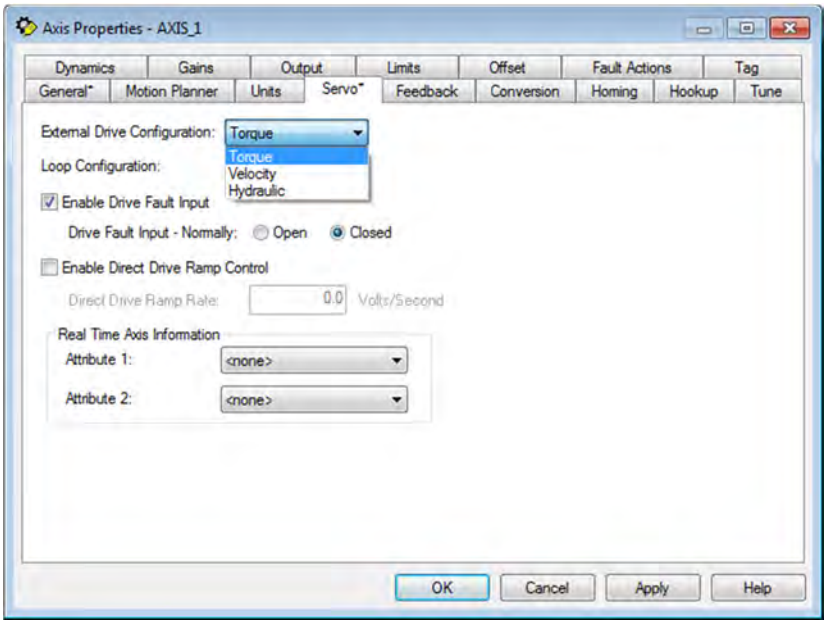
6. Click the **Browse** (**...**) button.

7. On the **Axis Properties** dialog box, in the **Module** box, choose the module to associate with the axis.



8. Click the **Servo** tab.

9. In the **External Drive Configuration** box, choose the drive configuration.



Tip: If you configure a torque drive then your drive must be able to be configured for torque. Hydraulic can only be selected if it is a hydraulic module.

10. Click **OK**.

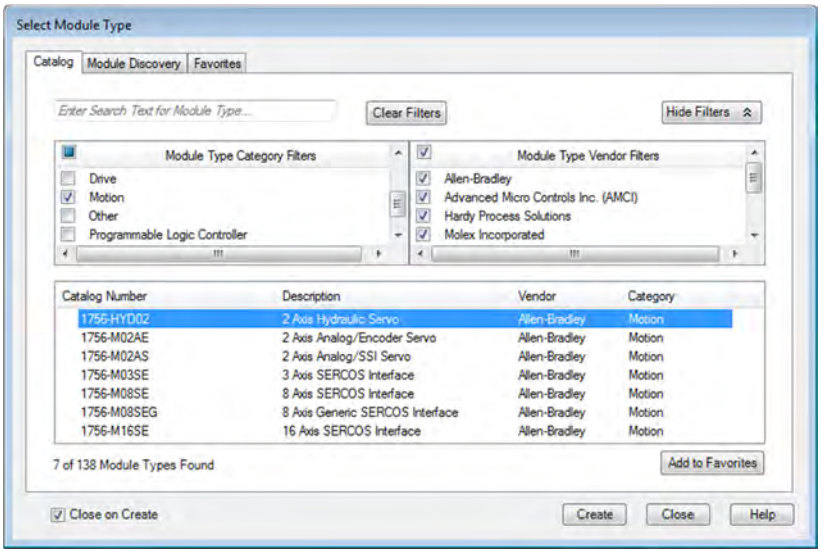
Add a hydraulic drive module

Follow these instructions to add a hydraulic drive module if your configuration includes it.

Important:	For all modules, use the firmware revision that goes with the firmware revision of your controller. See the release notes for your controller's firmware.

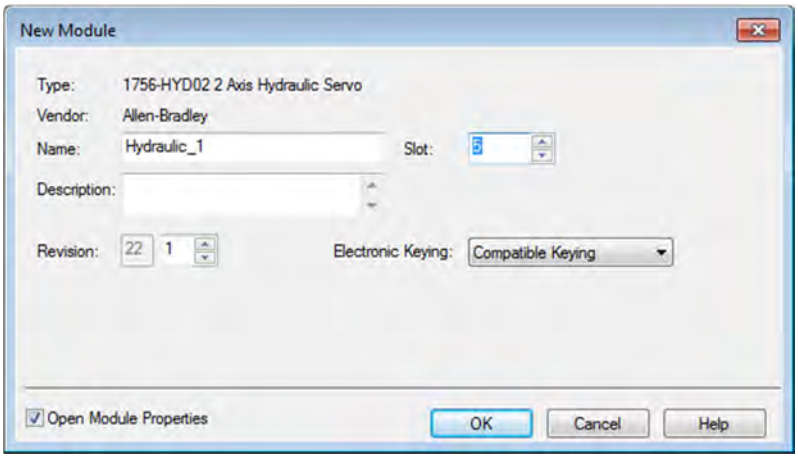
1. In the **Controller Organizer**, right-click the backplane and choose **New Module**.

2. On the **Select Module Type** dialog box, choose the hydraulic drive module that you want to add.



3. Select the **Close on Create** check box, and click **Create**.

4. On the **New Module** dialog box, in the **Name** box, type a name for the module.



5. In the **Slot** box, choose the number that corresponds to the physical slot that contains the module.

6. (optional) In the **Description** box, type a description.

7. In the **Electronic Keying** box, choose a keying option of either **Compatible Keying** or **Exact Match**. See Electronic Keying on page 26.

WARNING: Never select **Disable Keying** with motion modules.

8. Select the **Open Module Properties** check box.

9. Click **OK**. Continue with the instructions to modify the properties for the hydraulic drive module.

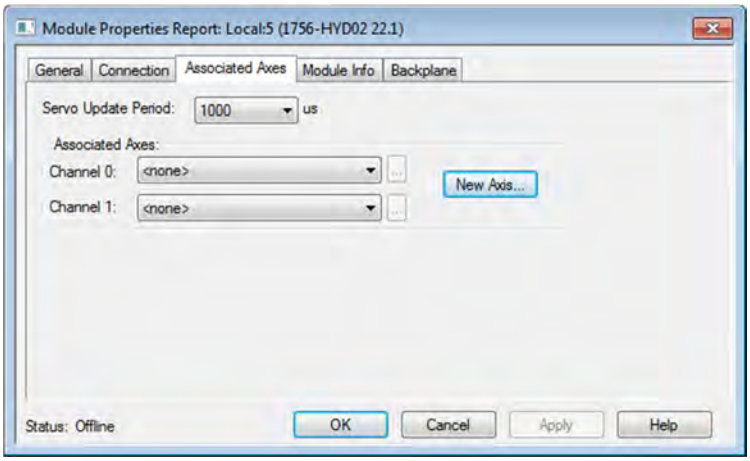
Modify properties for a hydraulic drive module

With a hydraulic drive, you must configure the feedback type. Based on the length of the feedback, the Servo Update Period must be configured. This setting is unique for the 1756-HYD02 module. If the Servo Update Period is not configured correctly, the axis does not work.

Follow these instructions to modify the properties for a hydraulic drive.

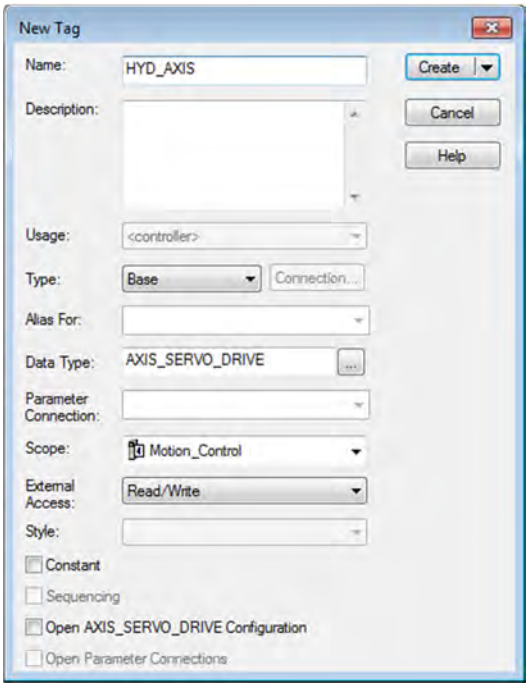
1. If the **Module Properties Report** dialog box is not already open, in the **Controller Organizer**, double-click the hydraulic drive module.

2. On the **Module Properties Report** dialog box, click the **Associated Axes** tab.



3. Click **New Axis** to create an AXIS_SERVO tag to associate to one of the channels.

4. On the **New Tag** dialog box, in the **Name** box, type a name for the axis tag.



5. Click **Create**.

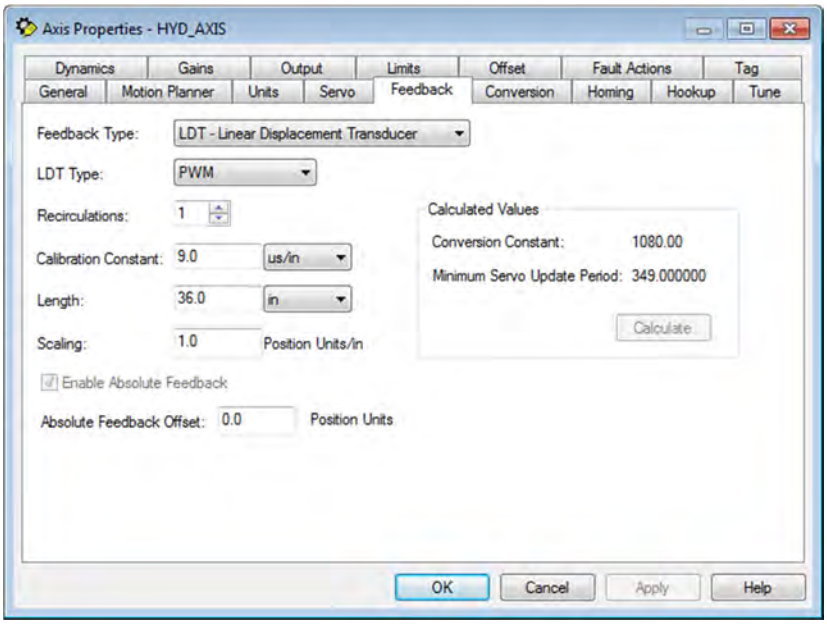
6. (optional) Repeat steps 3 through 5 if an additional axis is required.

7. On the **Module Properties Report** dialog box, in the **Channel 0** box, choose an axis.

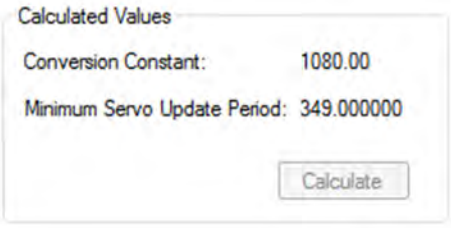
8. (optional) In the **Channel 1** box, choose an axis.
9. In the **Servo Update Period** box, select the periodic rate at which the module closes the servo loop for an axis.
10. Click **OK**.

Configure the feedback type

1. In the **Controller Organizer**, double-click the axis.
2. On the **Axis Properties** dialog box, click the **Feedback** tab.



3. In the **Feedback Type** box, choose the feedback type.
4. In the **Calibration Constant** box, choose the value and click **Calculate**. The minimum servo update period for the configured feedback appears.



5. If necessary, return to the **Module Properties** dialog box, and modify the settings on the **Associated Axis** tab.

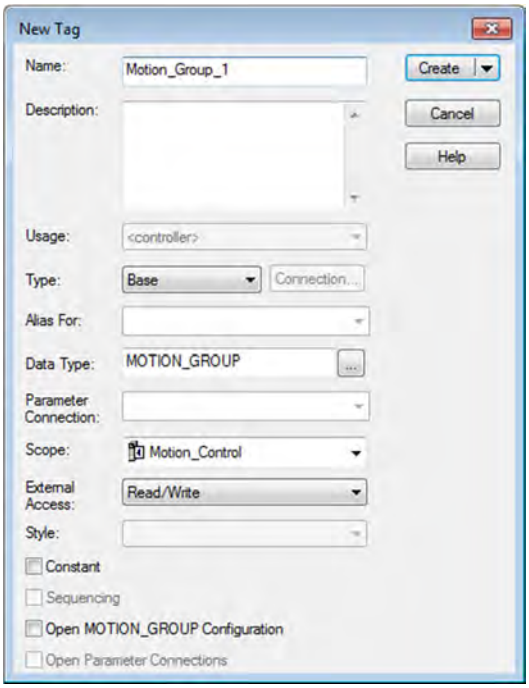
Add a motion group for Configure Analog Motion

Follow these instructions to add a motion group.

Important: Only one motion group can be created for each project.

1. In the **Controller Organizer**, right-click **Motion Groups** and choose **New Motion Group**.

2. On the **New Tag** dialog box, in the **Name** box, type a name for the motion group.



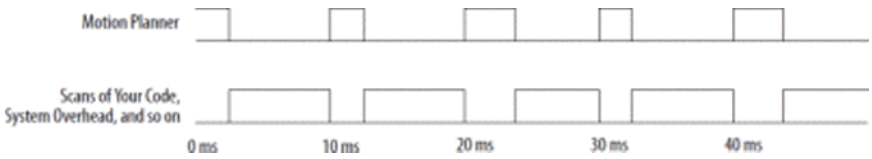
3. (optional) In the **Description** box, type a description.

4. Click **Create**.

Set the Base Update Period

The Base Update Period (also known as the Coarse Update Period) is how often the motion planner runs. The motion planner is the part of the controller that handles position and velocity information for the axes. When the motion planner runs, it interrupts most other tasks regardless of their priority.

Example: If the Base Update Period is set to 10 ms, then every 10 ms the controller stops scanning the code and performing other system overhead tasks, and runs the motion planner.

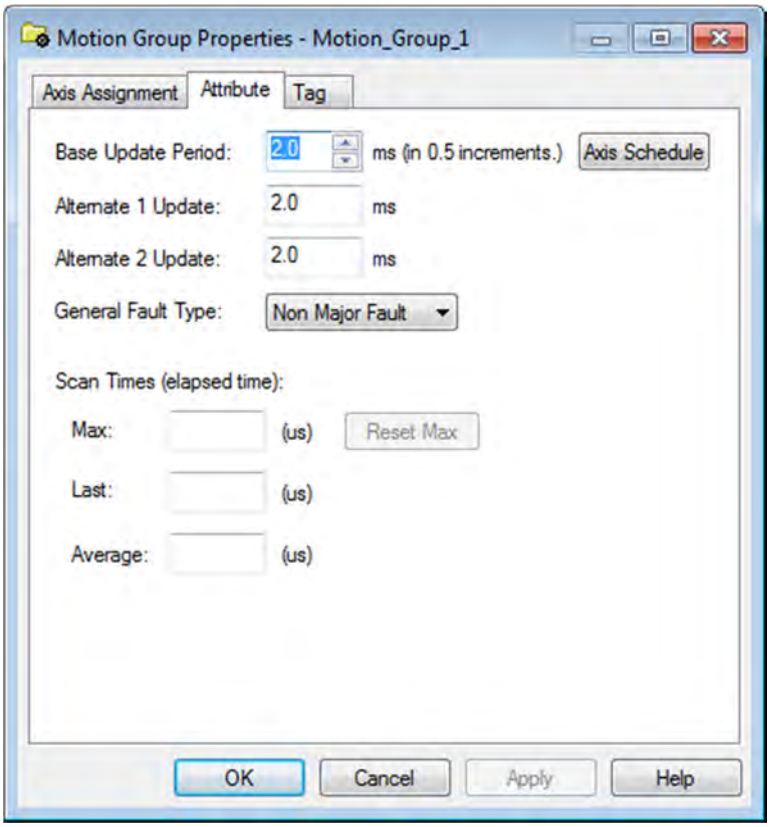


Use the following guidelines to set the Base Update Period.

Guideline	Description
Number of Axes	1756-L6*x* controller 4 axes/ms 1756-L7*x* controller8 axes/ms
Save Controller's Time	Leave at least half the controller's time for the scan of all your code.
Base Update Period and SERCOS modules	If you have SERCOS interface motion modules, set the Base Update Period to a multiple of the cycle time of the motion module. Example: If the cycle time is 2 ms, set the Base Update Period to 8 ms, 10 ms, 12ms, and so on.
Base Update Period and Analog modules	If you have analog motion modules, set the Base Update Period to: • at least 3 times the servo update period of the motion module. • a multiple of the servo update period of the motion module.

1. In the **Controller Organizer**, double-click the motion group.

2. Click the **Attribute** tab.



3. In the **Base Update Period** box, choose the update period using the guidelines mentioned earlier. The valid values range from **0.5** to **32**, in 0.5 increments.

   Tip: The **Axis Schedule** button opens the **Axis Schedule** dialog box where the base and alternate update periods can be scheduled and assigned to axes. Since axes used in coordinate system objects cannot be multiplexed, only the Base Update Period is used. Therefore, there is no need to open the **Axis Schedule** dialog box.

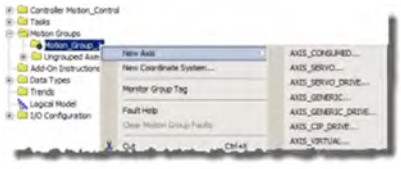
4. In the **General Fault Type** list, choose **Non Major Fault**.

5. Click **OK**.

Add an axis for Configure Analog Motion

Follow these instructions to add an axis for each of your drives.

1. In the **Controller Organizer**, right-click the motion group and choose **New Axis**.



2. Choose the data type based on the following guidelines.

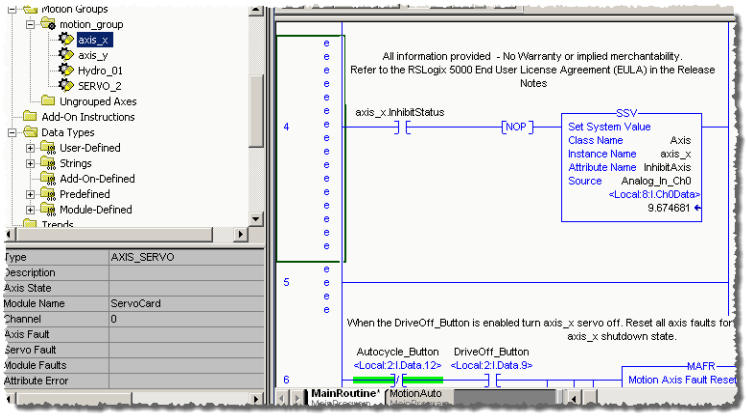
   - If you use one of the following motion modules, choose **AXIS_SERVO**.
     - 1756-M02AE
     - 1756-HYD02
     - 1756-M02AS
   - If you want to use a virtual configuration (no hardware), choose **AXIS_VIRTUAL**.

3. On the **New Tag** dialog box, in the **Name** box, type a name for the axis.
4. (optional) In the **Description** box, type a description for the axis.
5. Click **Create**.

Get Axis Information

You can get information about an axis in several ways.

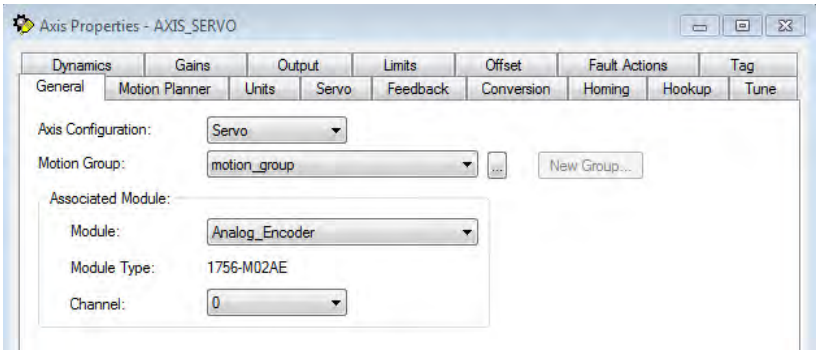
- Use the Quick View pane to see the state and faults of an axis.
- Use a Get System Value (GSV) instruction or Set System Value (SSV) instruction to read or change the configuration at run-time.
- Use the tag of the axis for status and faults.



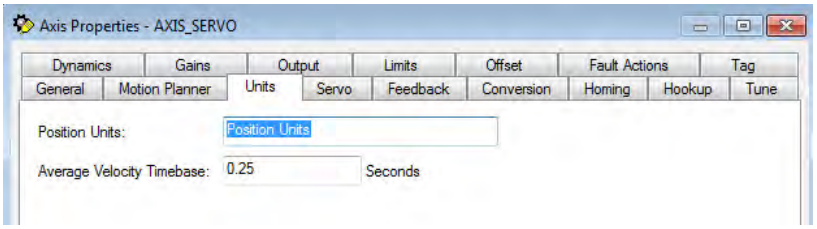
Configure an axis for Configure Analog Motion

Follow these steps to configure the axis of an analog module.

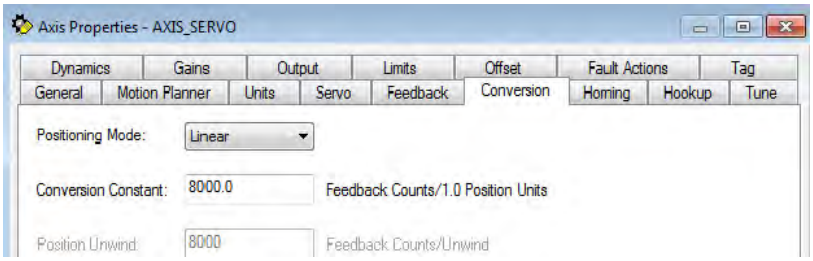
1. In the **Controller Organizer**, double-click the axis.
2. On the **General** tab, in the **Module** box, select the name that you gave to the drive for this axis.



3. Click the **Units** tab, and in the **Position Units** box, set the units in which you want to program.



4. Click the **Conversion** tab, and in the **Positioning Mode** and **Conversion Constant** boxes, enter the conversion details.

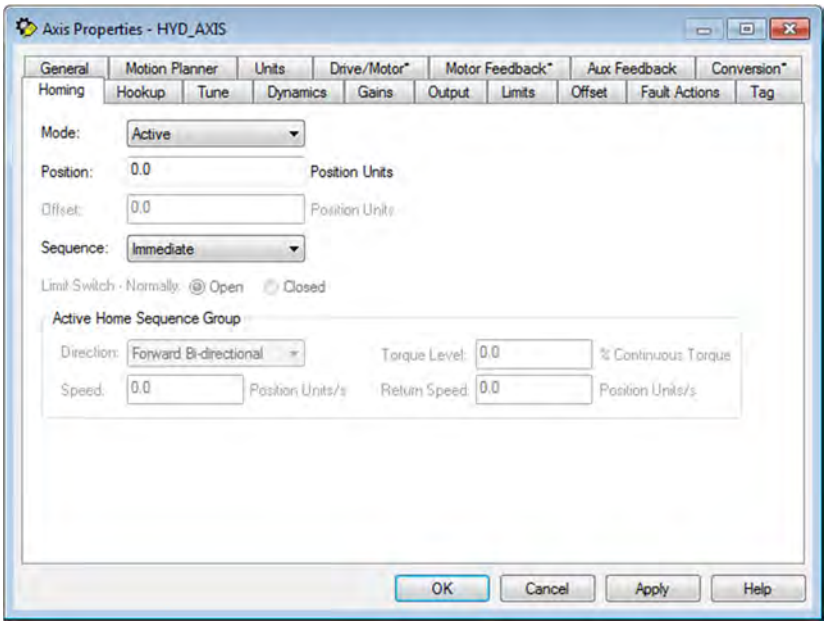


Set the homing sequence for Configure Analog Motion

For complete information about Homing modes, methods, and guidelines, see Home an Axis on page 111.

Follow these instructions to set the homing sequence.

1. Click the **Homing** tab, and in the **Mode** box, choose the homing mode.



2. In the **Position** box, type the position units.

3. In the **Sequence** box, choose the sequence type.

4. For all sequence types except **Intermediate**, in the **Description** box, choose the active home sequence type, and then in the **Speed** and **Return Speed** boxes, set the homing speeds.

5. Click **OK** to apply the changes.

Commission and tune

Introduction for Commission and Tune

This chapter discusses how to commission an axis for a motion application.

Download a program to the controller

Follow these instructions to download a program to the controller.

1. With the keyswitch, place the controller in Program or Remote Program mode.
2. From the **Communications** menu, choose **Download**.
3. Confirm that you wish to complete the download procedure.
4. Click **Download**.
5. When the download is complete, place the controller in Run/Test mode.

   After the project file is downloaded, status and compiler messages appear in the status bar.

Tip: When multiple workstations connect to the same controller using Logix Designer application and invoke the **Axis Wizard** or **Axis Properties** dialog box, the firmware only lets the first workstation make changes to axis attributes. The second workstation switches to a Read Only mode, indicated in the title bar, so that you may view the changes from that workstation, but not edit them.

If an axis in a motion group is open for edit, then any other workstation only gets read-only for any axis in that workstation, even if it is not the axis that the first workstation is modifying.

Test axis wiring and direction

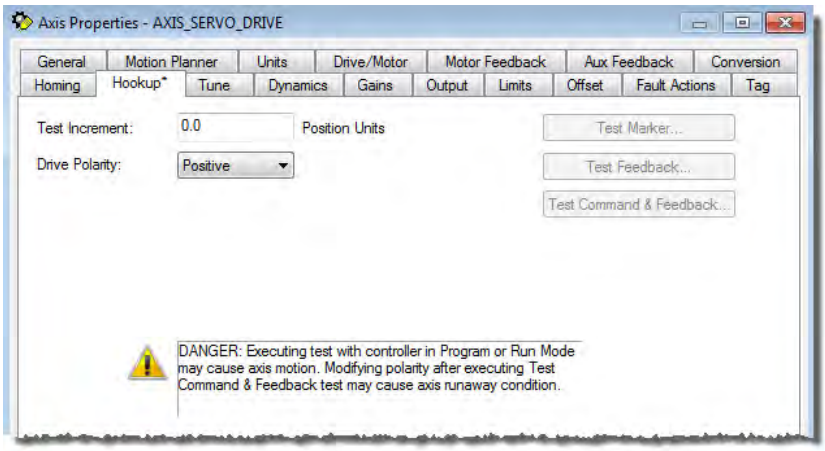
Follow these instructions to check the wiring of each drive.

Test	Description
Test marker	Checks that the drive gets the marker pulse. You must manually move the axis for this test.
Test feedback	Checks the polarity of the feedback. You must manually move the axis for this test.
Test command and feedback	Checks the polarity of the drive.

ATTENTION: These tests make the axis move even with the controller in remote program mode.

- Before you do the tests, make sure no one is in the way of the axis.
- **Do not** change the polarity after you do the tests. Otherwise you may cause an unexpected motion.

1. Download a program to the controller.
2. Place the controller in REM.
3. In the **Controller Organizer**, double-click the axis.
4. On the **Axis Properties** dialog box, click the **Hookup** tab.



Important: Make sure to follow all the dialogs or the information derived from the test is not saved to the axis configuration.

5. Click **Test Marker** and follow the additional dialog box instructions.
6. Click **Test Feedback** and follow the additional dialog box instructions.
7. Click **Test Command & Feedback** and follow the additional dialog box instructions.

Tune a SERCOS axis

Follow these instructions to tune an axis.

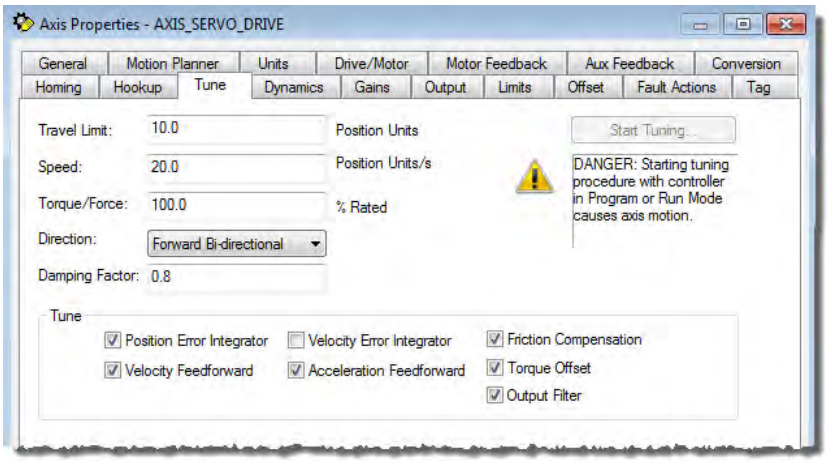
ATTENTION: When you tune an axis, it moves even with the controller in remote program mode. In that mode, your code is **not** in control of the axis.

ATTENTION: Before you tune an axis, make sure no one is in the way of the axis.

The default tuning procedure tunes the proportional gains. Typically, tune the proportional gains first and see how your equipment runs.

Tip: Where tighter positioning is required, Integral gain and feedforward constants can be selected. However, when used incorrectly, these settings can cause axis instability. See Tune on page 150.

1. Download a program to the controller.
2. Place the controller in REM.
3. In the **Controller Organizer**, double-click the axis.
4. Click the **Tune** tab.



5. In the **Travel Limit** box, type the limit of movement for the axis during the tuning procedure.
6. In the **Speed** box, type the maximum speed for your equipment.
7. Click **Start Tuning**.
8. Accept the changes to save the data derived from the tune as part of the axis configuration.

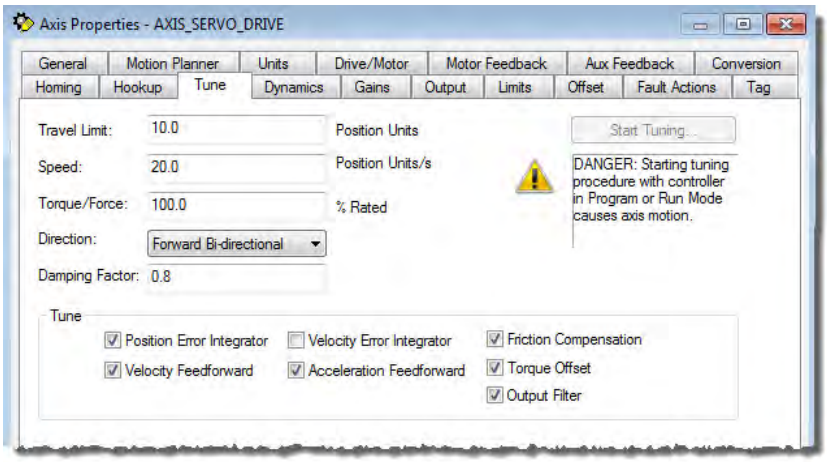
Tune an analog axis

Follow these instructions to tune an axis.

> When you tune an axis, it moves even with the controller in remote program mode. In that mode, your code is **not** in control of the axis. Before you tune an axis, make sure no one is in the way of the axis.

The default tuning procedure tunes the proportional gains. Typically, tune the proportional gains first and see how your equipment runs.

1. Download a program to the controller.
2. Place the controller in REM.
3. In the **Controller Organizer**, double-click the axis.
4. Click the **Tune** tab.



5. In the **Travel Limit** box, type the limit of movement for the axis during the tuning procedure.
6. In the **Speed** box, type the maximum speed for your equipment.
7. Click **Start Tuning**.

Troubleshoot faults

The following table explains the types of motion faults.

Type	Description	Example
Instruction error	Caused by a motion instruction: • Instruction errors do not impact controller operation. • Examine the error code in the motion control tag to see why an instruction has an error. • Fix instruction errors to optimize execution time and make sure that your code is accurate.	A Motion Axis Move (MAM) instruction with a parameter out of range

Type	Description	Example
Fault	Caused by a problem with the servo loop: • You choose whether motion faults give the controller major faults. • Can shut down the controller if you do not correct the fault condition.	• Loss of feedback • Actual position exceeding an overtravel limit

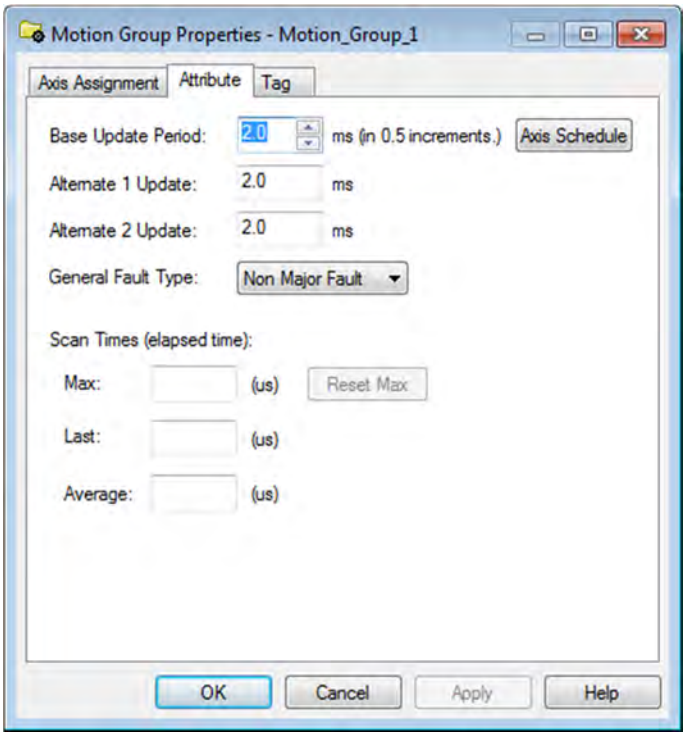
Manage motion faults

By default, the controller keeps running when there is a motion fault. As an option, you can have motion faults cause a major fault and shut down the controller.

1. Choose a Non-Major Fault.

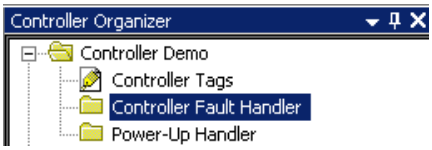
   Tip: If you select a Major Fault, then you should develop a Fault handler. See Logix5000 Controllers Major, Minor, and I/O Faults Programming Manual, publication 1756-PM014.

2. In the **Controller Organizer**, double-click the motion group.
3. Click the **Attribute** tab.
4. In the **General Fault Type** box, choose **Non Major Fault**.



5. Click **OK**.

6. In the **Controller Organizer**, drag your programs into the **Controller Fault Handler** folder so that the program runs when a fault occurs.



Configure the fault actions for an axis

Use the fault actions to set how an axis responds to faults. The type of faults depends on the type of axis and how you configure it.

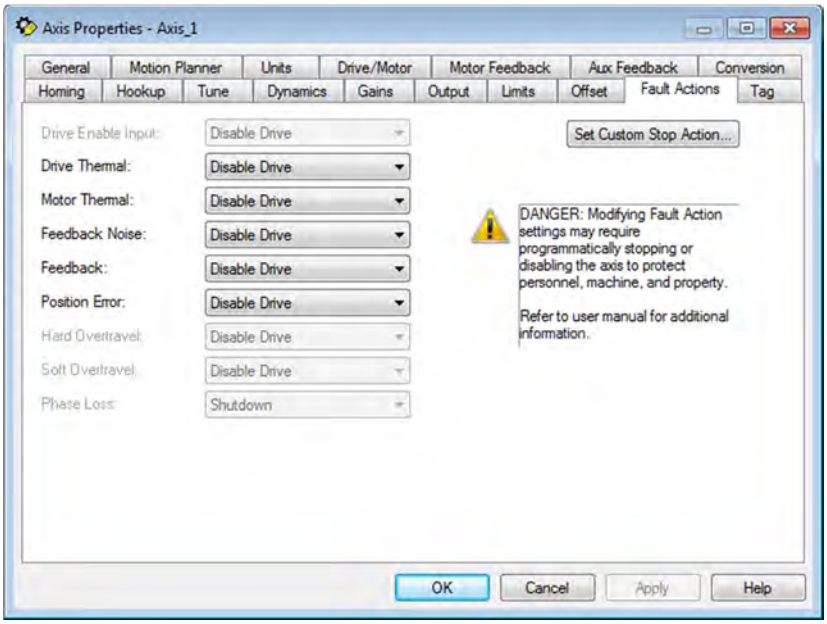
If you want to	Then choose	Description	
Shut down the axis and let it coast to a stop	Shutdown	Shutdown is the most severe action. Use it for faults that could endanger the machine or the operator if you do not remove power quickly and completely.	
		For this axis type	**When the fault happens**
		AXIS_SERVO	• Axis servo action is disabled. • The servo amplifier output is zeroed. • The drive enable output is deactivated. • The OK contact of the servo module opens. Use this to open the E-Stop string to the drive power supply. • This impacts both axes associated with the analog motion, not just the axis with the fault.
		AXIS_SERVO_DRIVE	• Axis servo action and drive power structure are immediately disabled. • The axis coasts to a stop unless you use some form of external braking.
Disable the axis and let the drive stop the axis using the best available stopping method	Disable Drive	**For this axis type**	**When the fault happens**
		AXIS_SERVO	• Planner decelerates axis motion to zero speed based on Maximum configured declaration using Trap Acc/Dec. • Axis servo action is off. • The servo amplifier output is zeroed. • The drive enable output is deactivated.
		AXIS_SERVO_DRIVE	• Planner decelerates axis motion to zero speed based on Maximum configured declaration using Trap Acc/Dec. • If the axis does not stop in the Stopping Time, the servo action and the power structure are disabled.
Leave the servo loop on and stop the axis at its Maximum Deceleration rate	Stop Motion	Use this fault action for less severe faults. It is the gentlest way to stop. Once the axis stops, you must clear the fault before you can move the axis.	
		For this axis type	**When the fault happens**
		AXIS_SERVO	The axis slows to a stop at the Maximum Deceleration Rate without disabling servo action or the servo module's Drive Enable output.
		AXIS_SERVO_DRIVE	• Control of the drive's servo loop is maintained. • The axis slows to a stop at the Maximum Deceleration rate without disabling the drive.

If you want to	Then choose	Description
Write your own application code to handle the fault	Status Only	Use this fault action only when the standard fault actions are not appropriate. With this fault action, you must write code to handle the motion faults. For Stop Motion or Status Only, the drive must stay enabled for the controller to continue to control the axis. Selecting Status Only only lets motion continue if the drive itself is still enabled and tracking the command reference.

Set the fault action for an axis

Use the following steps to set the fault actions for an axis.

1. In the **Controller Organizer**, double-click an axis.
2. Click the **Fault Actions** tab.



3. Set the desired attributes and click **OK**. (An analog axis will have fewer fault action selections than a SERCOS axis.)

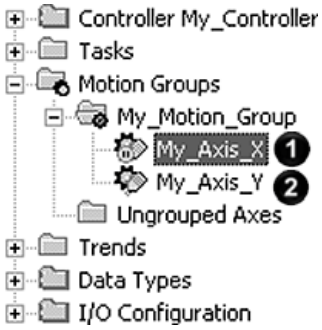
Inhibit an axis

Follow these instructions to block the controller from using an axis.

When to inhibit an axis

Use the following information to determine when to inhibit an axis.

Important:	Inhibiting an axis will take down ALL axes on the motion module or ring. The non-inhibited axes will then phase back up. Un-Inhibiting an axis will cause the same thing.



1. You want to block the controller from using an axis because the axis is faulted or not installed.
2. You want to let the controller use the other axes.

Example 1

If your equipment will need between 8 and 12 axes, depending on the application, create one project for all 12 axes. When you determine how many axes you need, inhibit the axes that you do not need.

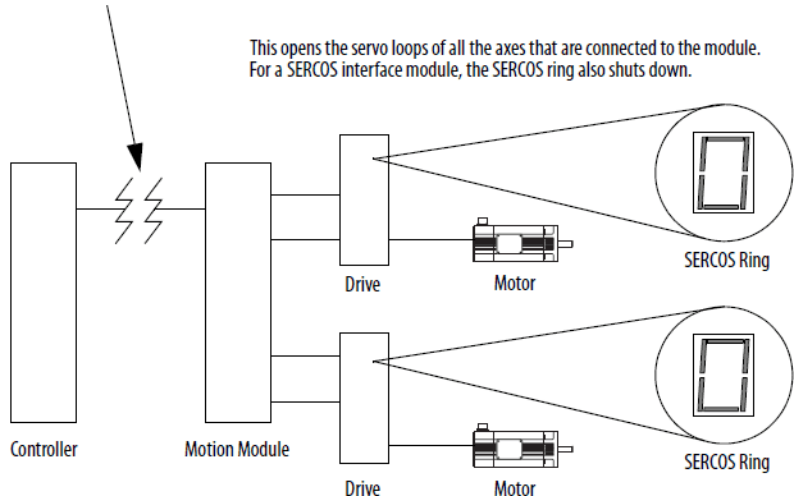
Example 2

If two production lines use the same SERCOS ring and one of the lines gets a fault, inhibit the axes on that line. This lets you run the other line while you take care of the fault.

Tip:	If an axis is faulted, all axes are still available. If there is a hardware issue with one of the "drives," you can inhibit the axis and remove the faulty hardware. When the ring phases back up, the inhibited axis (with its missing hardware) does prevent the rest of the axes from operating.

Before you begin

The following table explains what to do before you begin.

Before you inhibit or uninhibit an axis, turn off all axes.	Follow these steps before you inhibit or uninhibit an axis. 1. Stop all motion. 2. Open the servo loops of all axes. Use an instruction such as the Motion Servo Off (MSF) instruction. This lets you stop motion under your control. Otherwise the axes turn off on their own when you inhibit or uninhibit one of them. The connections to the motion module shut down when you inhibit or uninhibit an axis.  The controller automatically restarts the connections. The SERCOS ring also phases back up.
Inhibit only certain types of axes.	You can inhibit only the following types of axes. • AXIS_SERVO • AXIS_SERVO_DRIVE

To inhibit all axes of a motion module, inhibit the module instead.

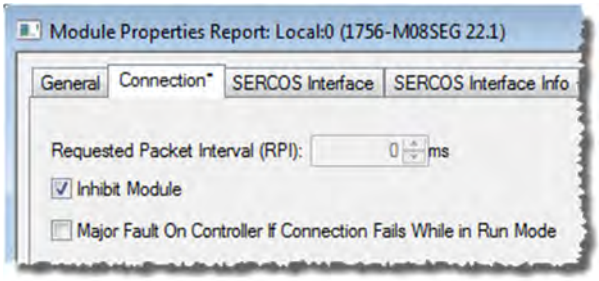
Do you want to inhibit all axes of a motion module?

- YES — Inhibit the motion module instead.
- NO — Inhibit the individual axes.

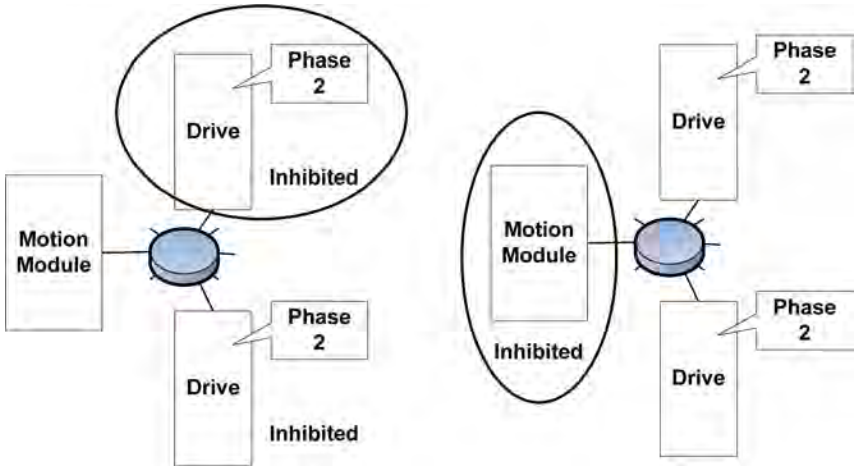
  You can inhibit all axes of a module on an individual basis. However, it is more efficient to inhibit all axes at once by inhibiting the module.

  Important: If you inhibit an axis on a drive, you inhibit all action on the drive, including any half axes. Make sure you are aware of all action on a drive before inhibiting the axis.

  Example: Suppose your motion module has two axes that you want to inhibit. In that case, just inhibit the module.

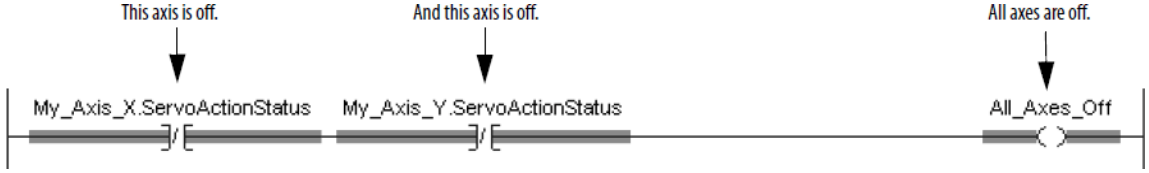


If you inhibit all axes on a SERCOS ring, the drives phase up to phase 2. This happens whether you inhibit each axis individually or you inhibit the motion module.

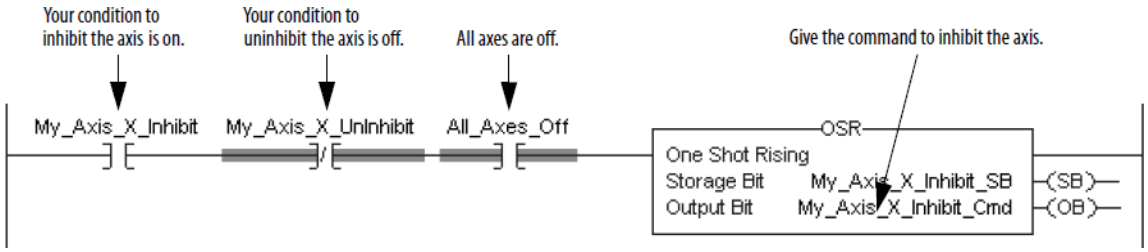


Example: Inhibit an axis

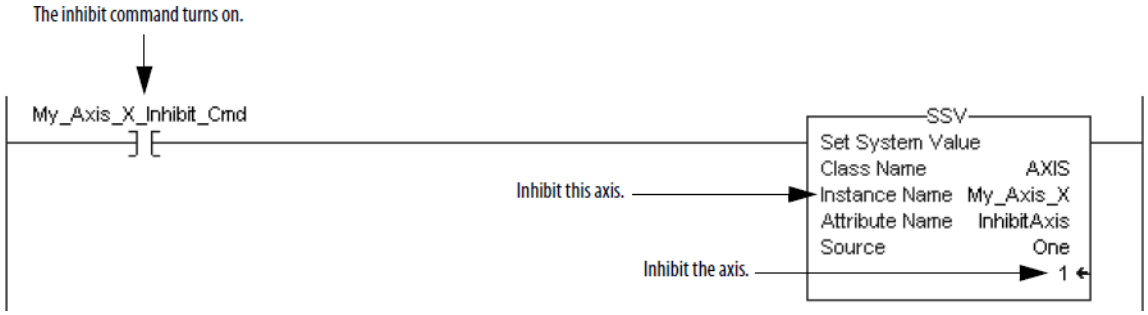
1. **Make sure all axes are off.**



1. **Use a one-shot instruction to trigger the inhibit.**



1. **Inhibit the axis.**



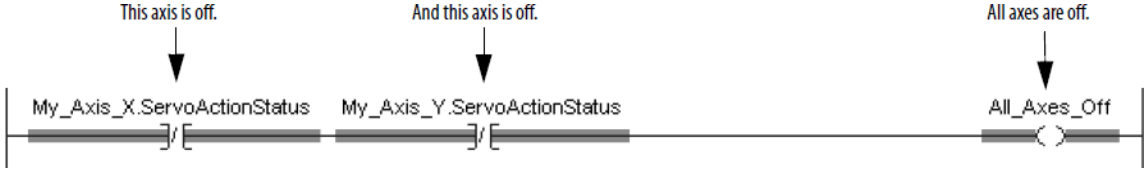
1. **Wait for the inhibit process to finish.**

All of these have happened:

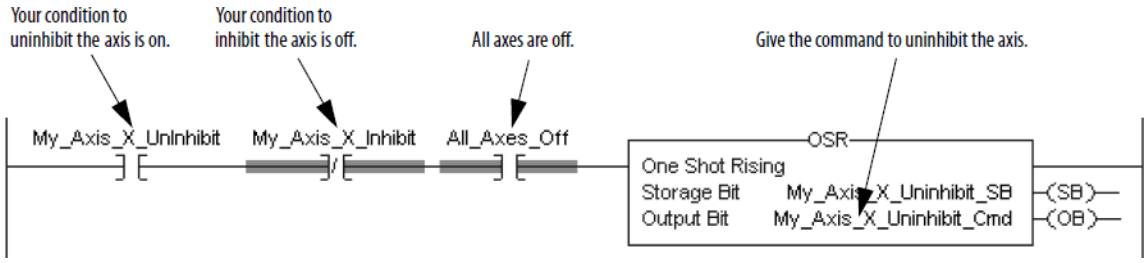
- The axis is inhibited.
- All uninhibited axes are ready.
- The connections to the motion module are running again.
- For a SERCOS ring, the SERCOS ring has phased up again.

Example: Uninhibit an axis

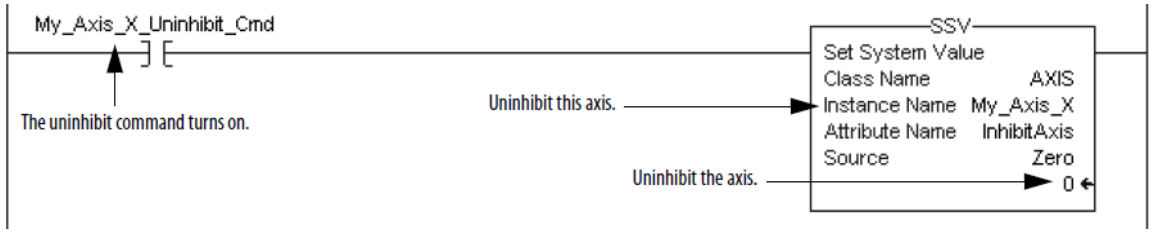
1. Make sure all axes are off.



1. **Use a one-shot instruction to trigger the uninhibit.**



1. **Uninhibit the axis.**



1. **Wait for the inhibit process to finish.**

   All of these have happened:

   - The axis is uninhibited.
   - All uninhibited axes are ready.
   - The connections to the motion module are running again.
   - For a SERCOS ring, the SERCOS ring has phased up again.

Test an axis with Motion Direct Commands

Motion Direct Commands let you issue motion commands while you are online without having to write or execute an application program.

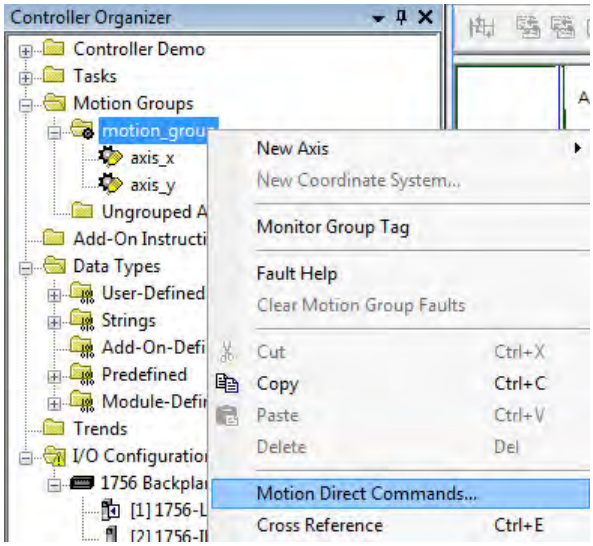
Motion Direct Commands are particularly useful when you are commissioning or debugging a motion application. During commissioning, you can configure an axis and monitor the behavior using Trends in the Controller Organizer. Use of Motion Direct Commands can fine-tune the system with or without load to optimize its performance. When in the testing and or debugging cycle, you can issue Motion Direct Commands to establish or reestablish conditions such as Home. Often during initial development or enhancement to mature applications you must test the system in small manageable areas. The tasks include the following.

- Home to establish initial conditions
- Incrementally Move to a physical position

- Monitor system dynamics under certain conditions

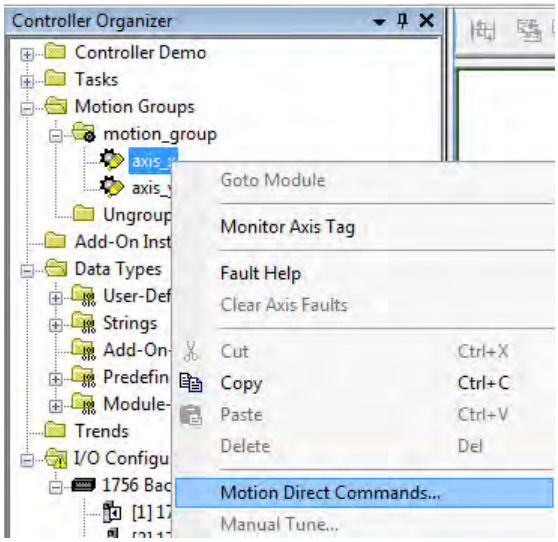
Access the Motion Direct Commands for a motion group

To access the Motion Direct Commands for a motion group, in the **Controller Organizer**, right-click the group.



Access the Motion Direct Commands for an axis

To access the Motion Direct Commands for an axis, in the **Controller Organizer**, right-click the axis.



Choose a command

Use the following table to choose an instruction and see if it is available as a Motion Direct Command.

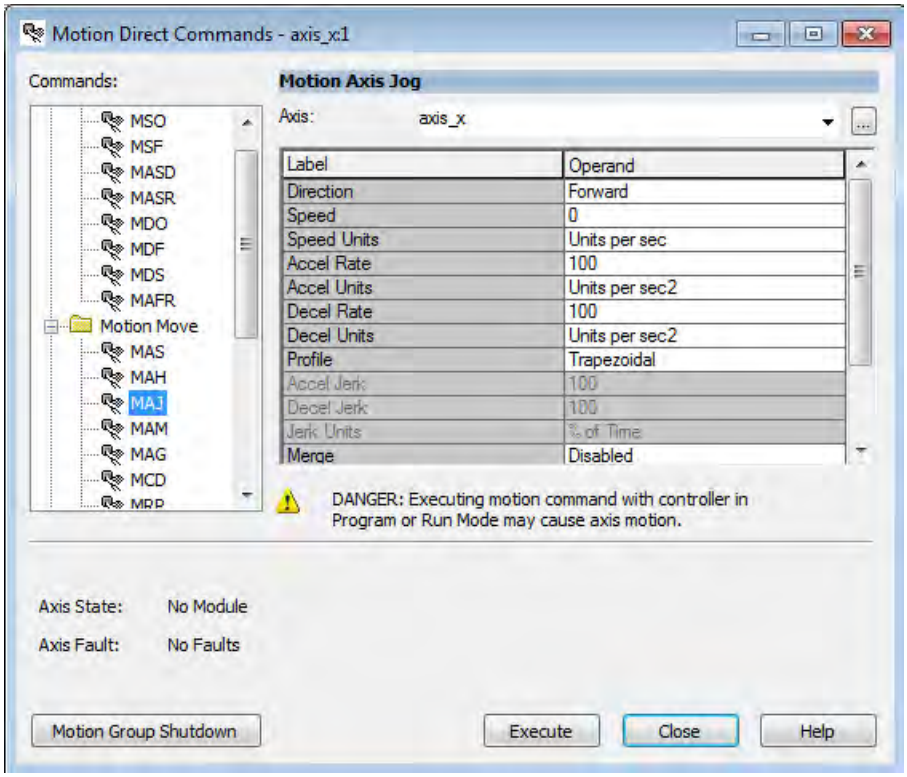
If you want to	And	Use this instruction	Motion Direct Command
Change the state of an axis	Enable the servo drive and activate the axis servo loop.	MSO Motion Servo On	Yes
	Turn off the servo drive and deactivate the axis servo loop.	MSF Motion Servo Off	Yes
	Force an axis into the shutdown state and block any instructions that initiate axis motion.	MASD Motion Axis Shutdown	Yes
	Transition an axis to the ready state. If all axes of a servo module are removed from the shutdown state as a result of this instruction, the OK relay contacts for only an analog module close.	MASR Motion Axis Shutdown Reset	Yes
	Enable the servo drive and set the servo output voltage of an axis.	MDO Motion Direct Drive On	Yes
	Turn off the servo drive and set the servo output voltage to the output offset voltage.	MDF Motion Direct Drive Off	Yes
	Clear all motion faults for an axis.	MAFR Motion Axis Fault Reset	Yes
If you want to	**And**	**Use this instruction**	**Motion Direct Command**
Control axis position	Stop any motion process on an axis.	MAS Motion Axis Stop	Yes
	Home an axis.	MAH Motion Axis Home	Yes
	Jog an axis.	MAJ Motion Axis Jog	Yes
	Move an axis to a position.	MAM Motion Axis Move	Yes
	Start electronic gearing between 2 axes.	MAG Motion Axis Gear	Yes
	Change the speed, acceleration, or deceleration of a move or a jog that is in progress.	MCD Motion Change Dynamics	Yes
	Define a Master/Slave relationship between two motion axes and select which type of move instructions.	MDAC Master Driven Axis Control	No
	Change the command or actual position of an axis.	MRP Motion Redefine Position	Yes
	Calculate a Cam Profile based on an array of cam points.	MCCP Motion Calculate Cam Profile	No
	Start electronic camming between 2 axes.	MAPC Motion Axis Position Cam	No
	Start electronic camming as a function of time.	MATC Motion Axis Time Cam	No

	Calculate the slave value, slope, and derivative of the slope for a cam profile and master value.	MCSV Motion Calculate Slave Values	No
Initiate action on all axes	Stop motion of all axes.	MGS Motion Group Stop	Yes
	Force all axes into the shutdown state.	MGSD Motion Group Shutdown	Yes
	Transition all axes to the ready state.	MGSR Motion Group Shutdown Reset	Yes
	Latch the current command and actual position of all axes.	MGSP Motion Group Strobe Position	Yes
Arm and disarm special event checking functions such as registration and watch position	Arm the watch-position event checking for an axis.	MAW Motion Arm Watch Position	Yes
	Disarm the watch-position event checking for an axis.	MDW Motion Disarm Watch Position	Yes
	Arm the servo-module registration-event checking for an axis.	MAR Motion Arm Registration	Yes
	Disarm the servo-module registration-event checking for an axis.	MDR Motion Disarm Registration	Yes
	Arm an output cam for an axis and output.	MAOC Motion Arm Output Cam	No
	Disarm one or all output cams connected to an axis.	MDOC Motion Disarm Output Cam	No
Tune an axis and run diagnostic tests for your control system. These tests include: • Motor/encoder hookup test • Encoder hookup test • Marker test	Use the results of an MAAT instruction to calculate and update the servo gains and dynamic limits of an axis.	MAAT Motion Apply Axis Tuning	No
	Run a tuning motion profile for an axis.	MRAT Motion Run Axis Tuning	No
	Use the results of an MRHD instruction to set encoder and servo polarities.	MAHD Motion Apply Hookup Diagnostic	No
	Run one of the diagnostic tests on an axis.	MRHD Motion Run Hookup Diagnostic	No
Control multi-axis coordinated motion	Start a linear coordinated move for the axes of coordinate system.	MCLM Motion Coordinated Linear Move	No
	Start a circular move for the axes of coordinate system.	MCCM Motion Coordinated Circular Move	No
	Change in path dynamics for the active motion on a coordinate system.	MCCD Motion Coordinated Change Dynamics	No
	Stop the axes of a coordinate system.	MCS Motion Coordinated Stop	No
	Shutdown the axes of a coordinate system.	MCSD Motion Coordinated Shutdown	No
	Transition the axes of a coordinate system to the ready state and clear the axis faults.	MCSR Motion Coordinated Shutdown Reset	No

	Start a transform that links two coordinate systems together.	MCT[1] Motion Coordinated Transform	No
	Calculate the position of one coordinate system with respect to another coordinate system.	MCTP[2] Motion Calculate Transform Position	No
	Define a Master/Slave relationship between a Master Axis and a Coordinate System.	MDCC Master Driven Coordinated Control	No

Motion Direct Command dialog box

You must be online to execute a Motion Direct Command, as the **Motion Group Shutdown** button and **Execute** button are only enabled when online. If you click one, action is taken immediately.



In the **Command** list, you can type the mnemonic and the list advances to the closest match or you can scroll down the list to select a command. Click the desired command and its dialog box opens.

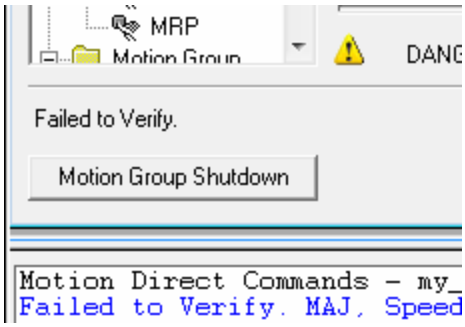
At the top of the dialog box, in the title bar, there is a number at the end of the axis or group that the command is being applied upon. This is the Instance reference

[1] You can use this instruction only with 1756-L6x controllers.
[2] You can use this instruction only with 1756-L6x controllers.

number. This number increases each time you open a new instance of the **Motion Direct Commands** dialog box. The number is cleared when you exit the Logix Designer application.

Motion Group Shutdown

The **Motion Group Shutdown** button is located to the left of the screen to avoid accidental invoking of this command when you really want to execute the command accessed from the Command tree.

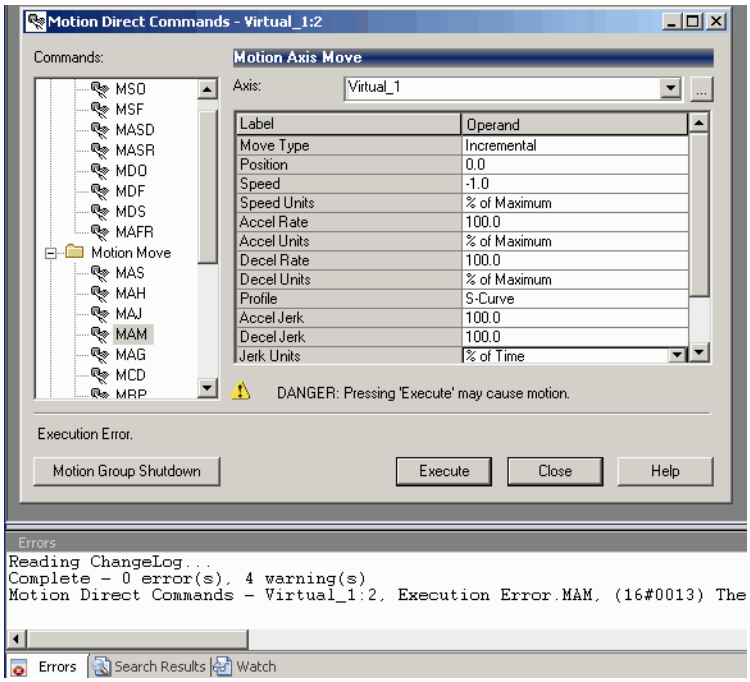


If you click **Motion Group Shutdown** and it is successfully executed, a Result message is displayed in the results window below the dialog box. Since the use of this Motion Group Shutdown is an abrupt means of stopping motion, an additional message is displayed in the error text field. The message 'MOTION GROUP SHUTDOWN executed!' is displayed to alert you that shutdown is complete. If the command fails, then an error is indicated as per normal operation.

Clicking **Execute** verifies the operands and initiates the current Motion Direct Command. There is space above **Motion Group Shutdown** and below the line where status text is displayed when a command is executed.

Motion Direct Command error process

When you execute a Motion Direct Command, there are two levels of error detections. The first level is verification of the command's operands. If a verification error is detected, a message 'Failed to Verify' is posted on the dialog box and a message is posted to the error result window. The second level is the initial motion direct command's error response return code. If an error code is detected, an 'Execution Error' message appears on the dialog box.

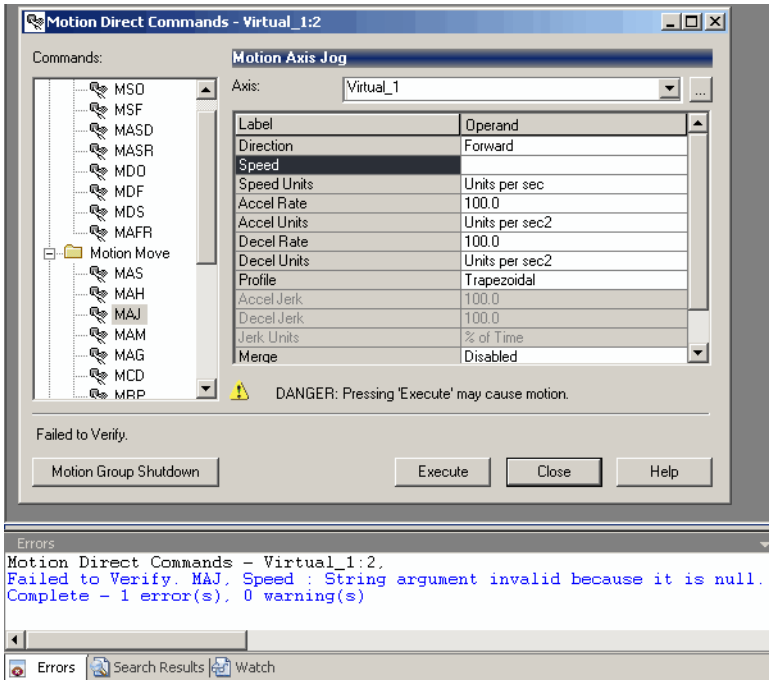


Regardless of whether an error is detected, a message appears in the Error result window describing the results of the executed command.

Motion Direct Command verification

When you select **Execute** from a **Motion Direct Command** dialog box, the operands are verified. If any operand fails verification, a 'Failed to Verify' message appears on the dialog box. A detailed error message appears in the error result window describing the fault.

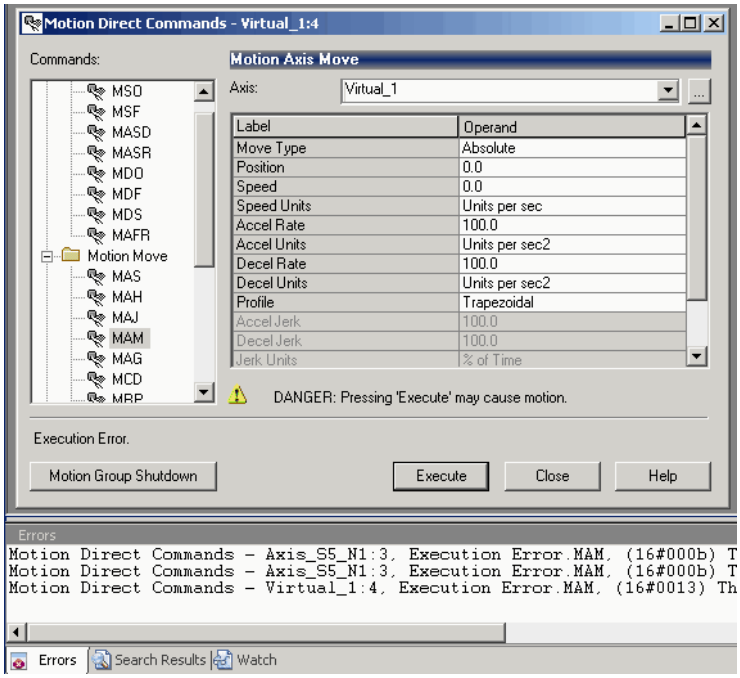
This allows multiple verification errors to display and provides navigation to the error source. That is, double-clicking the error in the Errors window opens the **Motion Direct Command** dialog box.



If no errors are detected, the status indicates which instruction was executed and that it had no errors.

Motion Direct Command execution error

When you select **Execute** from a **Motion Direct Command** dialog box and the operands are verified as valid, then the command is executed. If the command fails immediately, then an 'Execution Error' message appears on the dialog box. Regardless of whether an error is detected, a detailed message appears in the Error result window describing the immediate results of the executed command.



The message 'Execution Error' clears on subsequent command execution or if you select a new command. The information provided in the error result window after an execution is not cleared provides a history of what has been executed.

What if the software goes offline or the controller changes modes?

When the Logix Designer application transitions to Off-line and Direction Command instructions continue, if the controller Mode is changed, then the configured 'Programmed Stop Mode' controls what happens to motion.

Can two workstations give Motion Direct Commands?

When **Execute** is enabled and commands can be executed from a workstation, the group is locked. This means that another workstation cannot execute commands while this lock is in place. The lock is relinquished when all **Motion Direct Command** dialog boxes for the Motion Group are closed.

Program

Introduction

This chapter describes how to program a velocity profile and jerk rate.

You can use the following motion profiles for various instructions.

- **Trapezoidal** profile for linear acceleration and deceleration
- **S-curve** profiles for controlled jerk

Definition of Jerk

Jerk is the rate of change of acceleration or deceleration.

The jerk parameters only apply to S-curve profile moves using the following instructions.

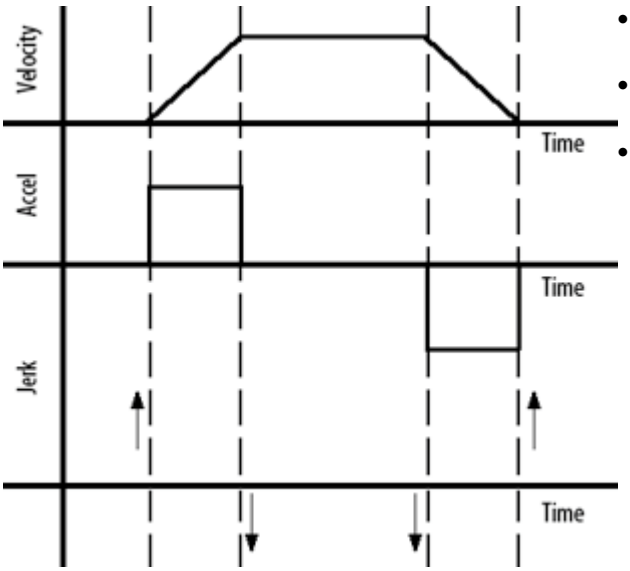
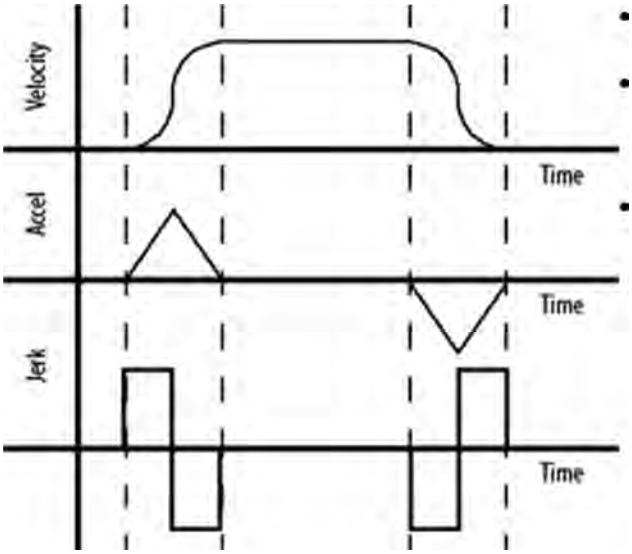
- MAJ
- MAM
- MAS
- MCD
- MCS
- MCCD
- MCCM
- MCLM

Example: If acceleration changes from 0 to 40 mm/s^2 in 0.2 seconds, the jerk is:

$(40\ mm/s^2 - 0\ mm/s^2) / 0.2\ s = 200\ mm/s^3$

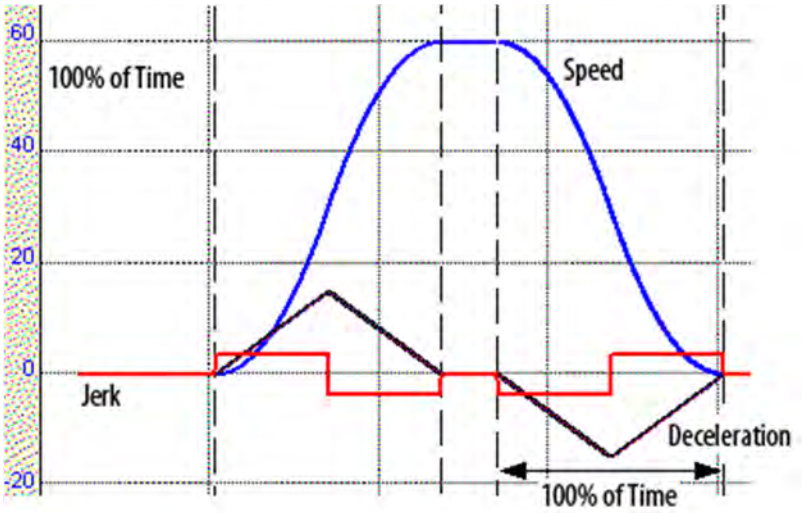
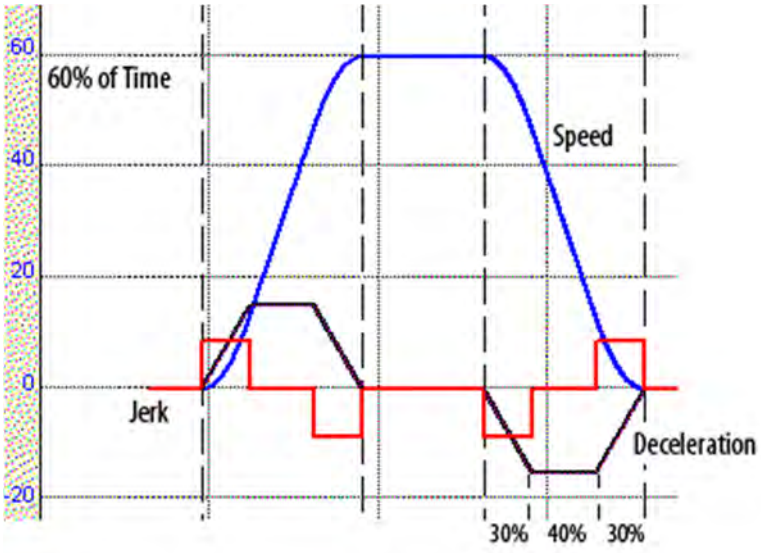
Choose a profile

Consider cycle time and smoothness when you choose a profile.

If you want	Choose this profile	Consideration
• Fastest acceleration and deceleration times • More flexibility in programming subsequent motion	Trapezoidal 	Jerk **does not** limit the acceleration and deceleration time: • The Acceleration and Deceleration rates control the maximum change in Velocity. • Your equipment and load get more stress than with an S-curve profile. • Jerk is considered infinite and is shown as a vertical line.
Smoother acceleration and deceleration that reduces the stress on the equipment and load	S-curve 	Jerk limits the acceleration and deceleration time. • More time is needed to accelerate and decelerate than a trapezoidal profile. • If the instruction uses an S-curve profile, the controller calculates acceleration, deceleration, and jerk when you start the instruction. • The controller calculates triangular acceleration and deceleration profiles.

Use % of Time for the easiest programming of jerk

Use % of Time to specify how much of the acceleration or deceleration time has jerk. You do not have to calculate actual jerk values.

Example	Profile
100% of Time	At 100% of Time, the acceleration or deceleration changes the entire time that the axis speeds up or slows down. 
60% of Time	At 60% of Time, the acceleration or deceleration changes 60% of the time that the axis speeds up or slows down. The acceleration or deceleration is constant for the other 40%. 

Velocity Profile Effects

This table summarizes the differences between profiles.

Profile	ACC/DEC	Motor	Priority of Control			
Type	Time	Stress	**Highest to Lowest**			
Trapezoidal	Fastest	Worst	Acc/Dec	Velocity	Position	
S-curve	2X Slower	Best	Jerk	Acc/Dec	Velocity	Position

Jerk Rate Calculation

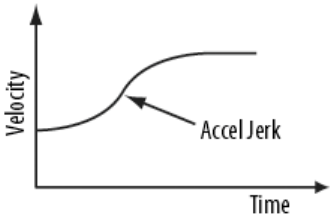
If the instruction uses or changes an S-curve profile, the controller calculates acceleration, deceleration, and jerk when you start the instruction.

The system has a Jerk priority planner. In other words, Jerk has a higher priority than acceleration and velocity. Therefore, you always get the programmed Jerk. If a move is velocity-limited, the move does not reach the programmed acceleration or velocity.

Jerk Parameters for MAJ programmed in units of % time are converted to engineering units as follows:

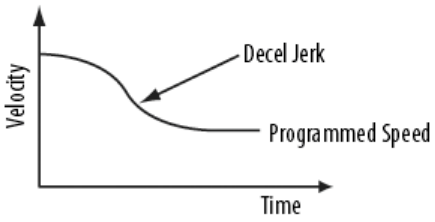
If Start Speed < MAJ Programmed Speed

$$\text{Accel Jerk (Units/Sec}^3\text{)} = \frac{\text{Programmed Accel Rate}^2}{\text{Programmed Speed}} * \left(\frac{200}{\text{\% of Time}} - 1 \right)$$



If Start Velocity > MAJ Programmed Speed

$$\text{Decel Jerk (Units/Sec}^3\text{)} = \frac{\text{Programmed Decel Rate}^2}{\text{Max (Programmed Speed, [Start Speed - Programmed Speed])}} * \left(\frac{200}{\text{\% of Time}} - 1 \right)$$

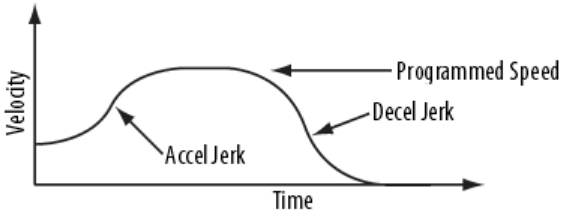


Jerks for programmed moves, such as MAM or MCLM instructions, in units of % time are converted to engineering units as follows:

If Start Speed < Programmed Speed

$$\text{Accel Jerk (Units/Sec}^3\text{)} = \frac{\text{Programmed Accel Rate}^2}{\text{Programmed Speed}} * \left(\frac{200}{\text{\% of Time}} - 1 \right)$$

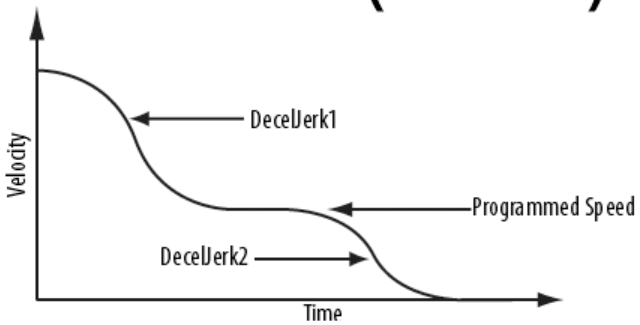
$$\text{Decel Jerk (Units/Sec}^3\text{)} = \frac{\text{Programmed Decel Rate}^2}{\text{Max (Programmed Speed, [Start Speed - Programmed Speed])}} * \left(\frac{200}{\text{\% of Time}} - 1 \right)$$



If Start Speed > Programmed Speed

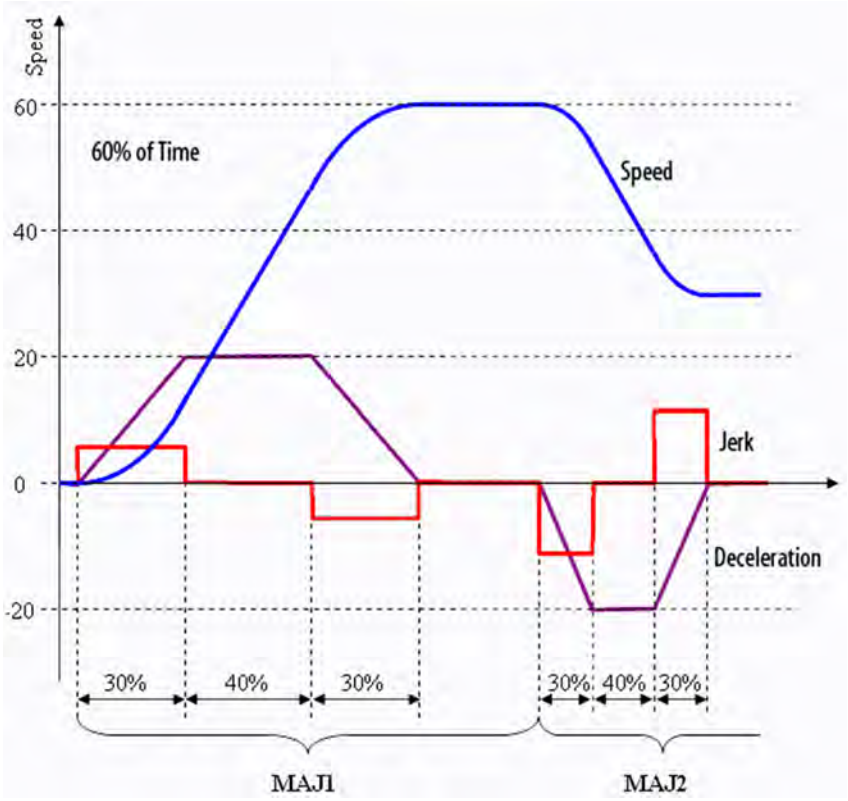
$$\text{DecelJerk1} = \frac{\text{Programmed Decel Rate}^2}{\text{Max (Programmed Speed, [Start Speed - Programmed Speed])}} * \left(\frac{200}{\text{\% of Time}} - 1 \right)$$

$$\text{DecelJerk2} = \frac{\text{Programmed Decel Rate}^2}{\text{Programmed Speed}} * \left(\frac{200}{\text{\% of Time}} - 1 \right)$$



DecelJerk1 is used while Current Speed > Programmed Speed
DecelJerk2 is used while Current Speed < Programmed Speed

Depending on the instruction's Speed parameter, the same '% of time jerk' can result in slopes for the acceleration profile that differ from the deceleration profile.



The motion planner algorithm adjusts the actual jerk rate so that the acceleration profile and the deceleration profile contains at least the '% of time' ramp time. If the Start Speed is close to the programmed Speed parameter, the actual percentage of ramp time may be higher than the programmed value.

In most cases the condition is:

if: (start Speed is == 0.0) OR (start Speed is > 2 * max Speed)

then: you get **programmed** percentage of ramp time

else: you get **higher than programmed** percentage of ramp time

Conversion from Engineering Units to % of Time

If you want to convert Engineering Units to % of Time, use these equations.

For Accel Jerk:

$$j_a[\%\ \text{of Time}] = \frac{2}{1 + \frac{j_a[EU/s^3]\, v_{max}[EU/s]}{a_{max}[EU/s^2]}} 100$$

For Decel Jerk:

$$j_d[\%\ \text{of Time}] = \frac{2}{1 + \frac{j_d[EU/s^3]\, v_{max}[EU/s]}{d_{max}[EU/s^2]}} 100$$

Jerk Programming in Units/Sec3

If you want to specify the jerk in 'Units/sec³' instead of '% of time', adjust your jerk value as follows so that you get the value that you programmed:

$$\text{Temporary Speed} = \frac{\text{Programmed Decel Rate}^2}{\text{Desired Decel Jerk value in Units/Sec}^3}$$

$$k = \frac{\text{Start Speed - Programmed Speed}}{\text{Max (Programmed Speed, Temporary Speed)}}$$

if (k < 1)
- Instruction faceplate Decel jerk in Units/Sec^3 = Desired Decel Jerk in Units/Sec^3

else
- Instruction faceplate Decel jerk in Units/Sec^3 = Desired Decel Jerk in Units/Sec^3 * k

Unique program considerations

If you program a move using the '% of Time units', Logix Designer application computes an:

Accel Jerk = a^2/v where a = the programmed Accel Rate and
v = programmed Speed.

Therefore, the higher the programmed speed, the lower the computed Jerk. The system has a Jerk priority planner.

In other words, Jerk has a higher priority than acceleration and velocity. Therefore, you always get the programmed Jerk. If a move is velocity-limited, the move does not reach the programmed acceleration and/or velocity.

Once you reach the velocity limit for the length of the move, as the velocity is increased, the move requires more and more time to complete.

'Decel Jerk' is computed similarly to the Accel Jerk described above. The only difference is that instead of a^2/v, Decel Jerk = d^2/v, where d = the programmed Decel Rate.

Example: **Example #1**

Start Speed = 8.0 in/sec

Desired Speed = 5.0 in/sec

Desired Decel Rate = 2.0 in/sec^2

Desired Decel Jerk = 1.0 in/sec^3

Temporary Speed = (Desired Decel Rate)2 / Desired jerk value in Units/Sec^3= 2.0^2 / 1.0 =
= 4.0 in/sec

k = (8.0 - 5.0) / max(5.0, 4.0) = 3.0 / 5.0 =
= 0.6

Because k < 1, we can enter the desired Decel jerk directly in the faceplate

Instruction faceplate Decel jerk in Units/Sec^3 = 1.0 in/sec^3

Example: **Example #2**

Start Speed = 13.0 in/sec

Desired Speed = 5.0 in/sec

Desired Decel Rate = 2.0 in/sec^2

Desired Decel Jerk = 1.0 in/sec^3

Temporary Speed = (Desired Decel Rate)^2 / Desired jerk value in Units/Sec^3= 2.0^2 / 1.0 =
= 4.0 in/sec

k = (13.0 - 5.0) / max(5.0, 4.0) = 8.0 / 5.0 =
= 1.6

Because k > 1, we have to calculate the Decel jerk to use on the instruction faceplate as:

Instruction faceplate Decel jerk in Units/Sec^3 =
= 1.0 in/sec^3 * 1.6 =
= 1.6 in/sec^3

Which revision do you have?

- 15 or earlier – % of Time is fixed at 100.
- 16 or later – % of Time defaults to 100% of time on projects converted from earlier versions. For new projects, you must enter the Jerk value.

Profile operand

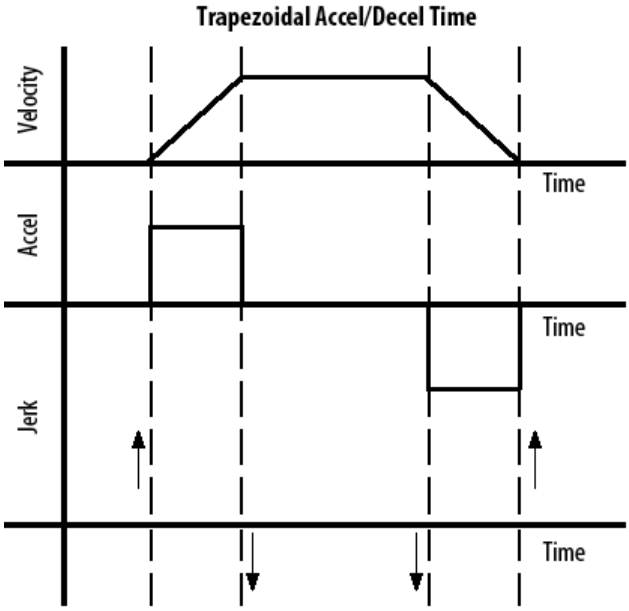
This operand has two profile types.

- Trapezoidal velocity profile

- S-Curve velocity profile

Trapezoidal velocity profile

The trapezoidal velocity profile is the most commonly used profile because it provides the most flexibility in programming subsequent motion and the fastest acceleration and deceleration times. The change in velocity per unit time is specified by acceleration and deceleration. Jerk is not a factor for trapezoidal profiles. Therefore, it is considered infinite and is shown as a vertical line in the following graph.



S-Curve velocity profile

S-Curve velocity profiles are most often used when the stress on the mechanical system and load needs to be minimized. The acceleration and deceleration time is balanced against the machine stress using two additional parameters, acceleration jerk and deceleration jerk.

Depending on the Jerk settings, the acceleration profile can be set to:

- Almost pure trapezoidal (fastest and highest stress).
- S-Curve (slowest, lowest stress).

The typical acceleration profile is a trade-off between stress and speed.

The Jerk is specified by the user (in Units/sec3 or as a percentage of maximum), or it is calculated from the percentage of time. (Percentage of time is equal to the percentage of ramp time in the acceleration/deceleration profile):

$$j_a\,[EU/s^3] = \frac{a_{max}{}^2\,[EU/s^2]}{v_{max}\,[EU/s]} \left(\frac{200}{j_a\,[\%\ of\ time]} - 1 \right)$$

$$j_a\,[EU/s^3] = \frac{d_{max}{}^2\,[EU/s^2]}{v_{max}\,[EU/s]} \left(\frac{200}{j_a\,[\%\ of\ time]} - 1 \right)$$

Backward compatibility

The Jerk of 100% of time produces triangular acceleration and deceleration profiles.

Very small Jerk rates that are less than 5% of time, produce acceleration and deceleration profiles close to trapezoidal ones.

Important:	Higher values of the % of Time result in lower values of Jerk Rate Limits and, therefore, slower profiles. See the following table for reference.

	Trapezoidal Velocity Profile[3]	**S-shaped Velocity Profile with 1< = Jerk <100% of Time**[4]	**S-shaped Velocity Profile with Jerk = 100% of Time**[5]

[3] The example labeled Trapezoidal Accel/Decel Time uses a rectangular acceleration profile.

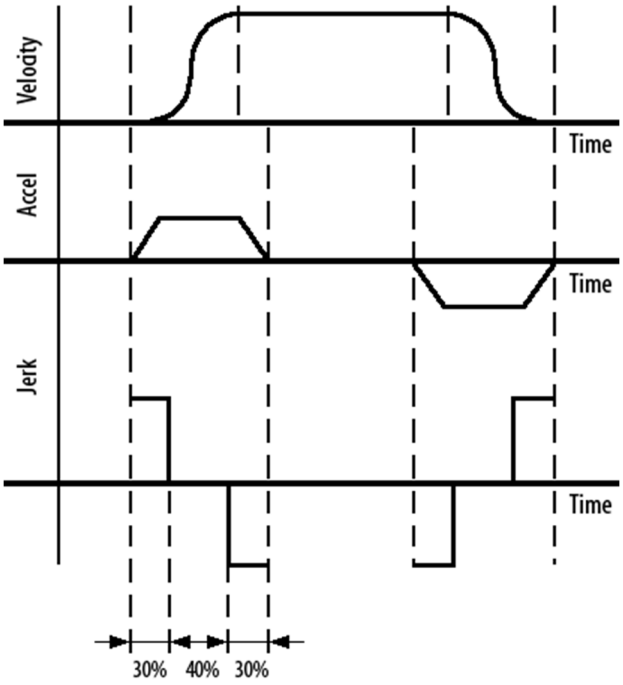
[4] The example labeled Programmable S-Curve Accel/Decel Time, Acceleration Jerk = 60% of Time uses a trapezoidal acceleration profile.

[5] The example labeled S-Curve Accel/Decel Time, Backward Compatibility Setting: Acceleration Jerk = 100% of Time uses a triangular acceleration profile.

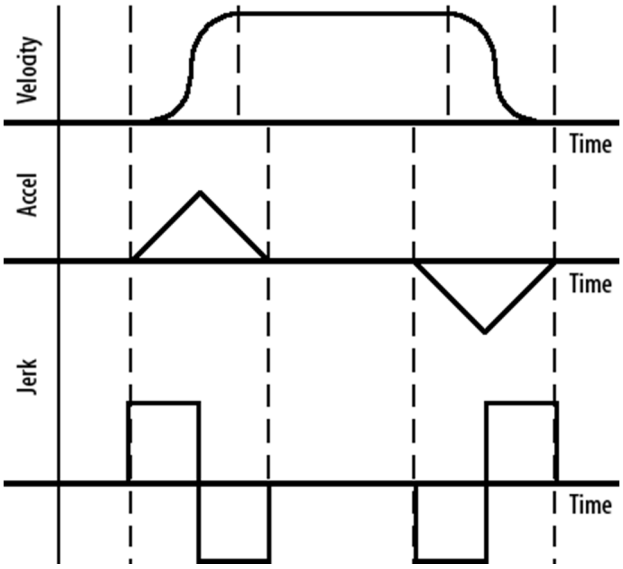
Accel/Decel Jerk in Units/sec3	∞	$\frac{\text{Max Accel}^2}{\text{Max Velocity}}$ to ∞	$\frac{\text{Max Accel}^2}{\text{Max Velocity}}$
Accel/Decel Jerk in % of Maximum	NA	0 - 100%	NA
Accel/Decel Jerk in % of Time	0%	1 - 100%	100%

Calculations are performed when an Axis Move, Change Dynamics, or an MCS Stop of StopType = Move or Jog is initiated.

Programmable S-Curve Accel/Decel Time Acceleration Jerk = 60% of Time



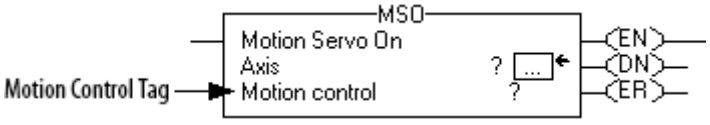
S-Curve Accel/Decel Time, Backward Compatibility Setting: Acceleration Jerk = 100% of Time



Enter basic logic

The controller gives you a set of motion control instructions for your axes.

- Uses the instructions just like the rest of the Logix5000 instructions. You can program motion control in the following programming languages.
  - Ladder diagram (LD)
  - Structured text (ST)
  - Sequential function chart (SFC)
- Each motion instruction works on one or more axes.
- Each motion instruction needs a motion control tag. The tag uses a MOTION_INSTRUCTION data type. The tag stores the status information of the instruction.



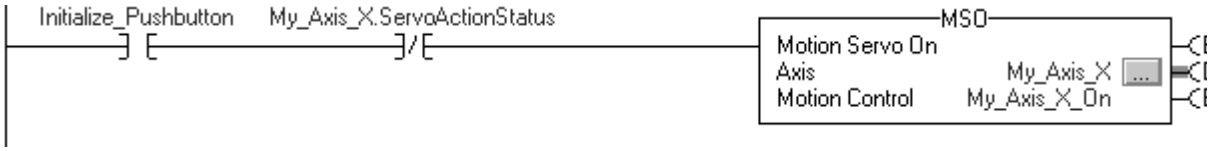
ATTENTION: Use the tag for the motion control operand of motion instruction only once. Unintended operation of the control variables may happen if you re-use the same motion control tag in other instructions.

Example: Motion control program

This is an example of ladder logic that homes, jogs, and moves an axis.

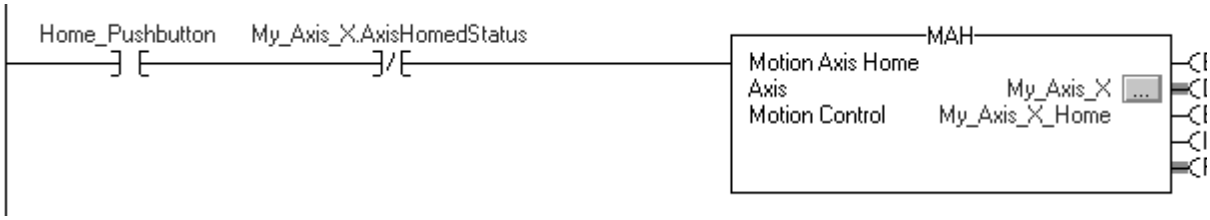
If *Initialize_Pushbutton* = on and the axis = off (*My_Axis_X.ServoActionStatus* = off) then

The MSO instruction turns on the axis.



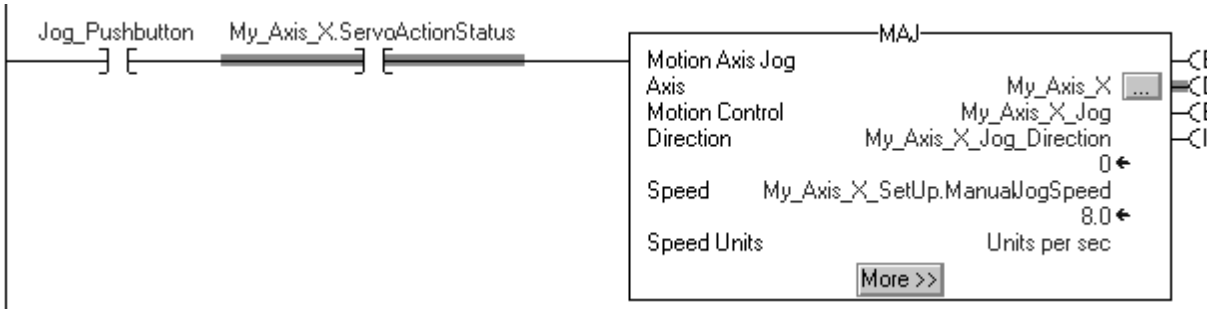
If *Home_Pushbutton* = on and the axis has not been homed (*My_Axis_X.AxisHomedStatus* = off) then

The MAH instruction homes the axis.



If *Jog_Pushbutton* = on and the axis = on (*My_Axis_X.ServoActionStatus* = on) then

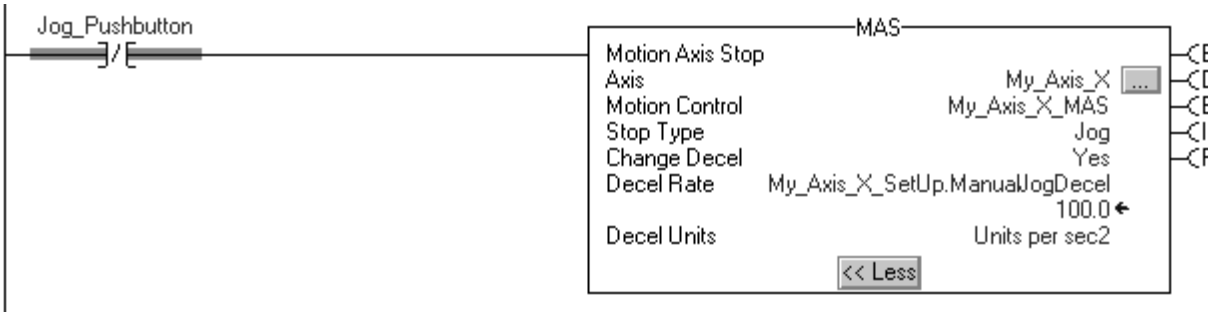
The MAJ instruction jogs the axis forward at 8 units/s.



If *Jog_Pushbutton* = off then

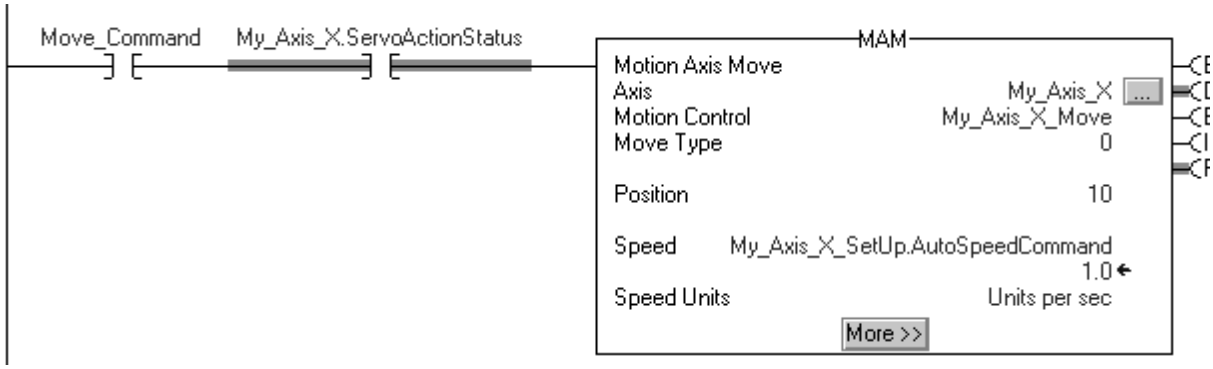
The MAS instruction stops the axis at 100 units/s^2

Make sure that *Change Decel* is *Yes*. Otherwise, the axis decelerates at its maximum speed.



If *Move_Command* = on and the axis = on (*My_Axis_X.ServoActionStatus* = on) then

The MAM instruction moves the axis. The axis moves to the position of 10 units at 1 unit/s.



Download a program and run the logic

Follow these instructions to download your program to a controller.

1. With the keyswitch, place the controller in Program or Remote Program mode.
2. From the **Communications** menu, choose **Download**.
3. Confirm that you wish to complete the download procedure.
4. Click **Download**.
5. Once the download is complete, place the controller in Run/Test mode.

   After the project file is downloaded, status and compiler messages appear in the status bar.

Choose a motion instruction

Use the following table to choose an instruction and see if it is available as a Motion Direct Command.

If You Want To	And	Use This Instruction	Motion Direct Command
Change the state of an axis	Enable the drive and activate the axis servo loop.	MSO Motion Servo On	Yes
	Disable the drive and deactivate the axis servo loop.	MSF Motion Servo Off	Yes
	Force an axis into the shutdown state and block any instructions that initiate axis motion.	MASD Motion Axis Shutdown	Yes
	Reset the axis from the shutdown state.	MASR Motion Axis Shutdown Reset	Yes

If You Want To	And	Use This Instruction	Motion Direct Command
	Enable the drive and set the servo output voltage of an axis.	MDO Motion Direct Drive On	Yes
	Disable the drive and set the servo output voltage to the output offset voltage.	MDF Motion Direct Drive Off	Yes
	Clear all motion faults for an axis.	MAFR Motion Axis Fault Reset	Yes
Control axis position	Stop any motion process on an axis.	MAS Motion Axis Stop	Yes
	Home an axis.	MAH Motion Axis Home	Yes
	Jog an axis.	MAJ Motion Axis Jog	Yes
	Move an axis to a position.	MAM Motion Axis Move	Yes
	Start electronic gearing between 2 axes.	MAG Motion Axis Gear	Yes
	Change the speed, acceleration, or deceleration of a move or a jog that is in progress.	MCD Motion Change Dynamics	Yes
	Define a Master/Slave relationship between two motion axes and select the type of move instructions.	MDAC Master Driven Axis Control	No
	Change the command or actual position of an axis.	MRP Motion Redefine Position	Yes
	Calculate a Cam Profile based on an array of cam points.	MCCP Motion Calculate Cam Profile	No
	Start electronic camming between 2 axes.	MAPC Motion Axis Position Cam	No
	Start electronic camming as a function of time.	MATC Motion Axis Time Cam	No
	Calculate the slave value, slope, and derivative of the slope for a cam profile and master value.	MCSV Motion Calculate Slave Values	No
Initiate action on all axes	Stop motion of all axes.	MGS Motion Group Stop	Yes
	Force all axes into the shutdown state.	MGSD Motion Group Shutdown	Yes
	Transition all axes to the ready state.	MGSR Motion Group Shutdown Reset	Yes
	Latch the current command and actual position of all axes.	MGSP Motion Group Strobe Position	Yes
Arm and disarm special event checking functions such as registration and watch position	Arm the watch-position event checking for an axis.	MAW Motion Arm Watch Position	Yes
	Disarm the watch-position event checking for an axis.	MDW Motion Disarm Watch Position	Yes

If You Want To	And	Use This Instruction	Motion Direct Command
	Arm the servo-module registration-event checking for an axis.	MAR Motion Arm Registration	Yes
	Disarm the servo-module registration-event checking for an axis.	MDR Motion Disarm Registration	Yes
	Arm an output cam for an axis and output.	MAOC Motion Arm Output Cam	No
	Disarm one or all output cams connected to an axis.	MDOC Motion Disarm Output Cam	No
Tune an axis and run diagnostic tests for your control system. These tests include: • Motor/encoder hookup test • Encoder hookup test • Marker test	Uses the results of an MRAT instruction to calculate and update the servo gains and dynamic limits of an axis.	MAAT Motion Apply Axis Tuning	No
	Run a tuning motion profile for an axis.	MRAT Motion Run Axis Tuning	No
	Use the results of an MRHD instruction to set encoder and servo polarities.	MAHD Motion Apply Hookup Diagnostic	No
	Run one of the diagnostic tests on an axis.	MRHD Motion Run Hookup Diagnostic	No
Control multi-axis coordinated motion	Start a linear coordinated move for the axes of coordinate system.	MCLM Motion Coordinated Linear Move	No
	Start a circular move for the axes of coordinate system.	MCCM Motion Coordinated Circular Move	No
	Change in path dynamics for the active motion on a coordinate system.	MCCD Motion Coordinated Change Dynamics	No
	Stop the axes of a coordinate system or cancel a transform.	MCS Motion Coordinated Stop	No
	Shutdown the axes of a coordinate system.	MCSD Motion Coordinated Shutdown	No
	Start a transform that links two coordinate systems together. This is like bi-directional gearing.	MCT Motion Coordinated Transform[6]	No
	Calculate the position of one coordinate system with respect to another coordinate system.	MCTP Motion Calculate Transform Position[7]	No
	Transition the axes of a coordinate system to the ready state and clear the axis faults.	MCSR Motion Coordinated Shutdown Reset	No
	Define a Master/Slave relationship between a Master Axis and a Coordinate System.	MDCC Master Driven Coordinated Control	No

Sample projects

To access the sample projects, take the following steps.

[6] You can use this instruction only with 1756-L6x controllers.
[7] You can use this instruction only with 1756-L6x controllers.

1. On the **Help** menu, click **Quick Start**.
2. On the **Quick Start** window, in the left navigation pane, expand **Controller Projects**, and click **Open Sample Project**.
3. In the **Open Project** dialog box, click **MSG_To_multiple_Controllers.acd**, and click **Open**.

Free sample code is also available at http://samplecode.rockwellautomation.com.

Troubleshoot axis motion

The following are situations that could happen while you are running an axis, and how to troubleshoot them.

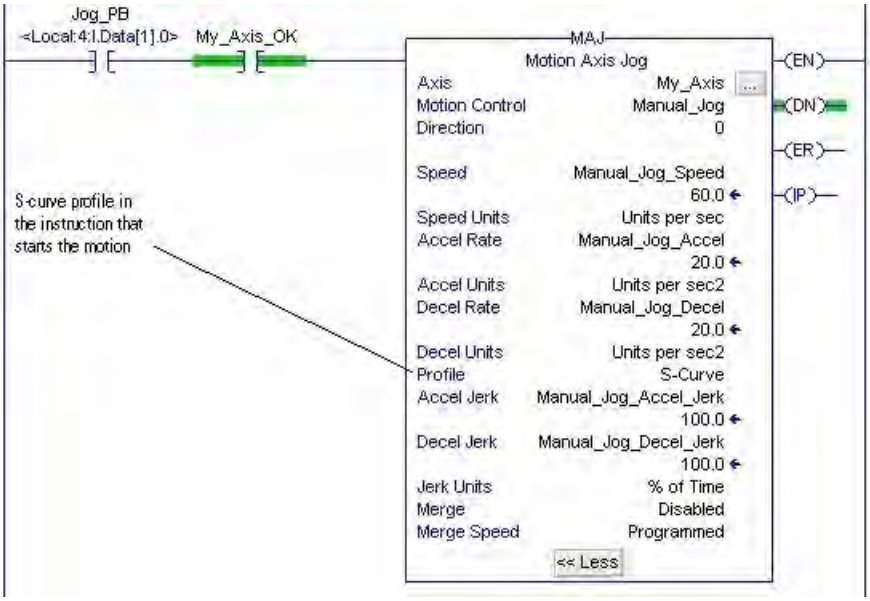
Why does my axis accelerate when I stop it?

While an axis is accelerating, you try to stop it. The axis keeps accelerating for a short time before it starts to decelerate.

Example

You start a Motion Axis Jog (MAJ) instruction. Before the axis gets to its target speed, you start a Motion Axis Stop (MAS) instruction. The axis continues to speed up and then eventually slows to a stop.

Look For



Cause

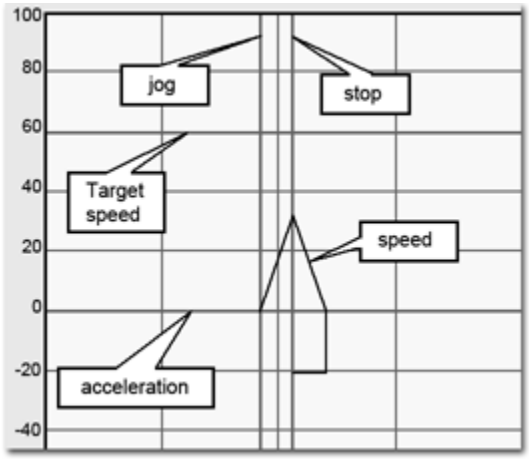
When you use an S-curve profile, jerk determines the acceleration and deceleration time of the axis.

- An S-curve profile must get acceleration to 0 before the axis can slow down.
- The time required depends on the acceleration and speed.
- In the meantime, the axis continues to speed up.

The following trends show how the axis stops with a trapezoidal profile and an S-curve profile.

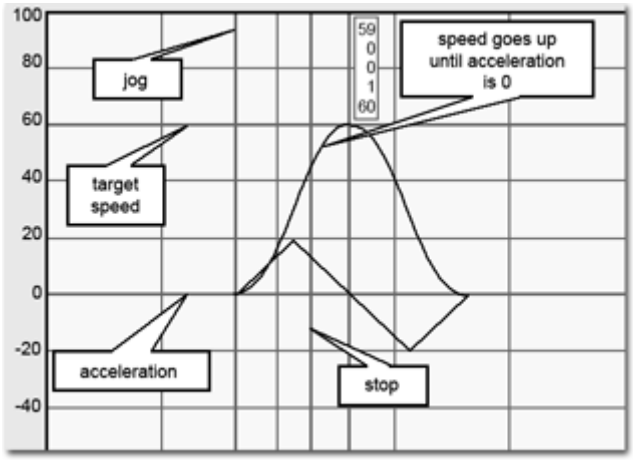
Stop while accelerating

Trapezoidal



The axis slows down as soon as you start the stopping instruction.

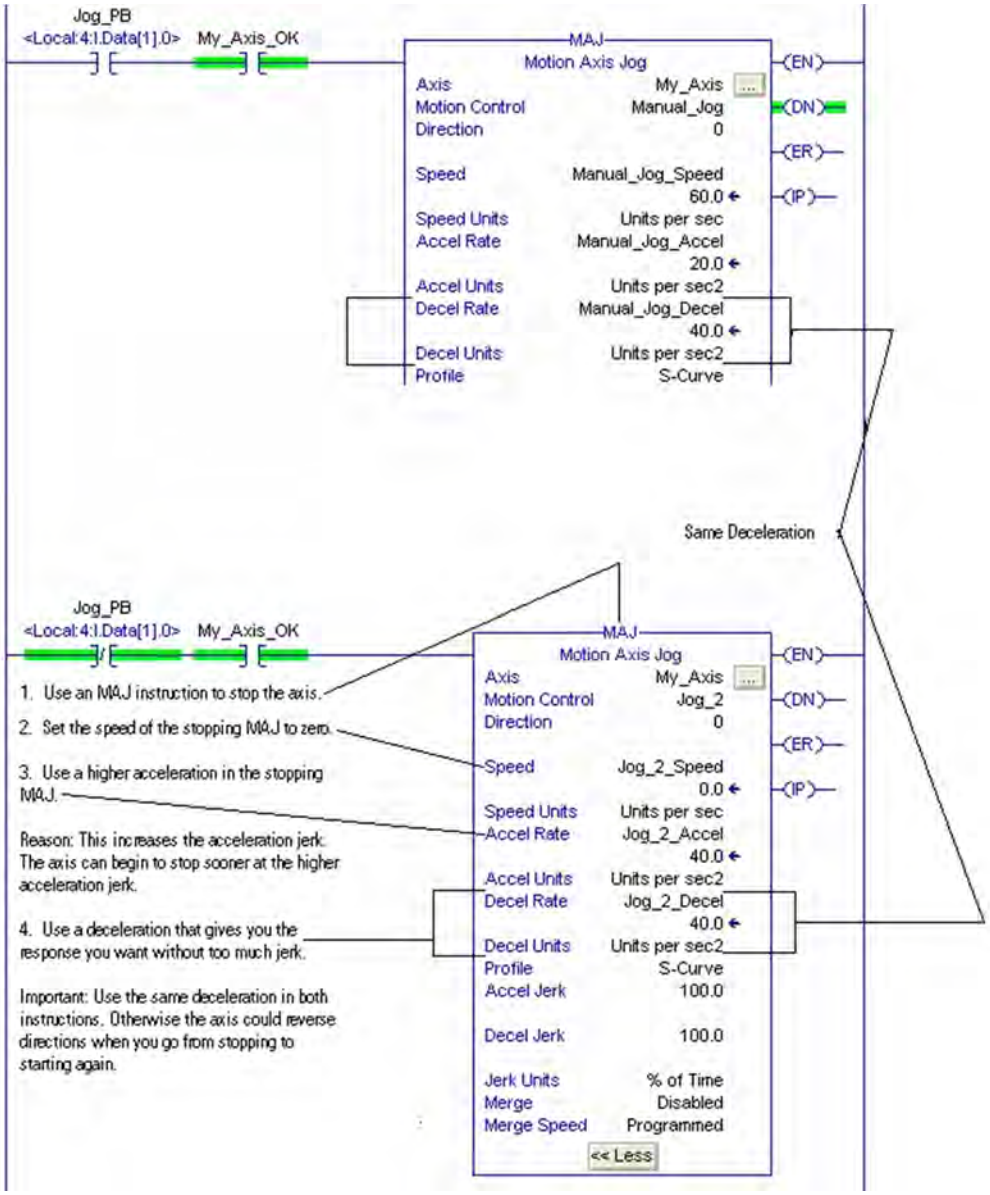
S-curve



The axis continues to speed up until the S-curve profile brings the acceleration rate to 0.

Corrective Action

Revision 15 or earlier



Revision 16 or later

Leave bit 0 of the DynamicsConfigurationBits attribute for the axis turned ON. Otherwise, this corrective action will not work.
For more information. Use the following path:
Help >Index>GSV/SSV Objects>Axis>Dynamics Configuration Bits

Revision 16 and later lets you increase the deceleration jerk of an Motion Action Stop (MAS) instruction to get a quicker stop.

If the Jerk Units Are	Then Make This Change to the Decel Jerk
% of Time	Reduce the % of Time on the Decel Jerk
% of Maximum	Increase the % of Maximum on the Decel Jerk
Units per sec^3	Increase Units per sec^3 on the Decel Jerk

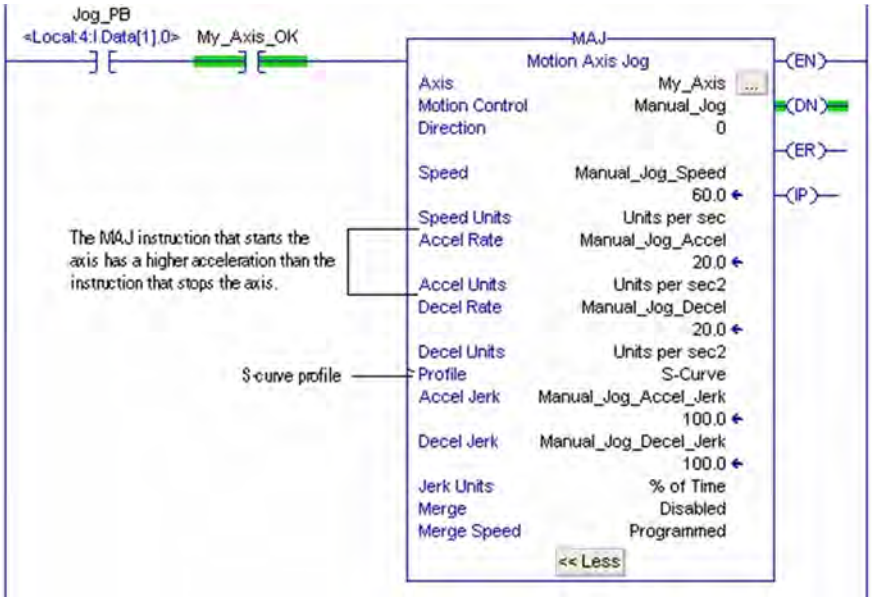
Why does my axis overshoot its target speed?

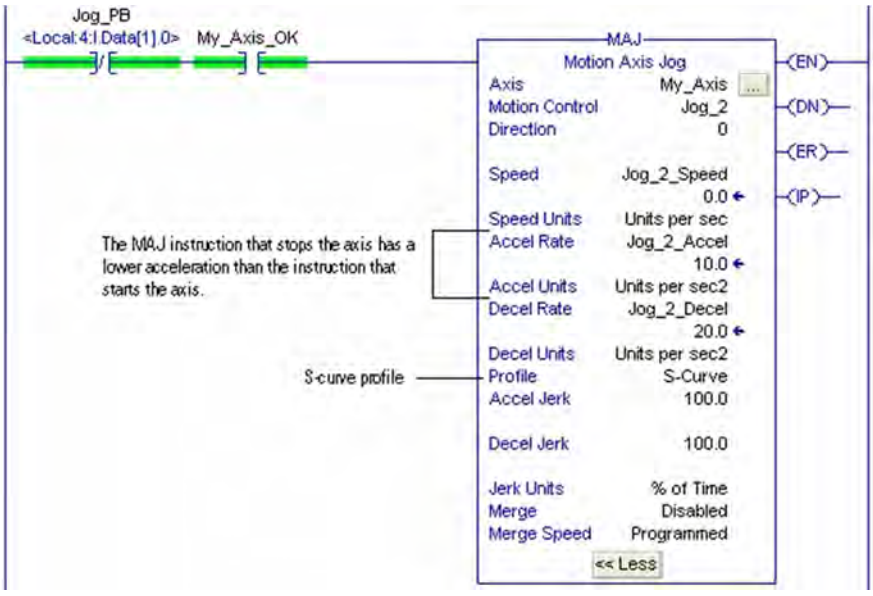
While an axis is accelerating, you try to stop the axis or change its speed. The axis keeps accelerating and goes past its initial target speed. Eventually it starts to decelerate.

Example

You start a Motion Axis Jog (MAJ) instruction. Before the axis gets to its target speed, you try to stop it with another MAJ instruction. The speed of the second instruction is set to 0. The axis continues to speed up and overshoots its initial target speed. Eventually it slows to a stop.

Look For





Cause

When you use an S-curve profile, jerk determines the acceleration and deceleration time of the axis.

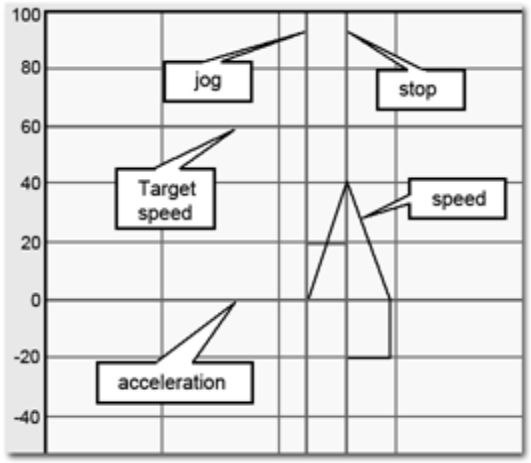
- An S-curve profile must get acceleration to 0 before the axis can slow down.

- If you reduce the acceleration, more time is required to get acceleration to 0.
- In the meantime, the axis continues past its initial target speed.

The following trends show how the axis stops with a trapezoidal profile and an S-curve profile.

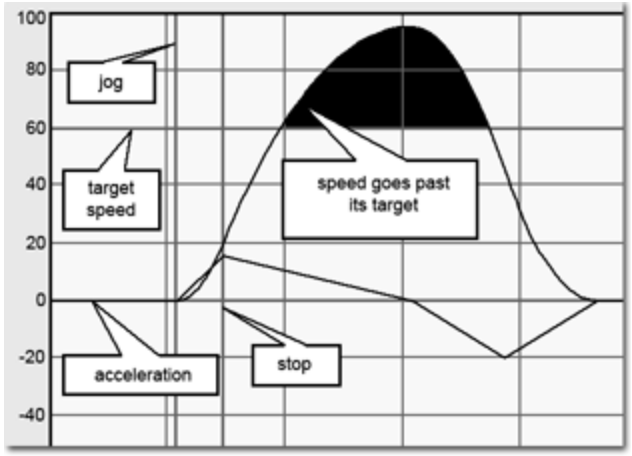
Stop while accelerating and reduce the acceleration rate

Trapezoidal



The axis slows down as soon as you start the stopping instruction. The lower acceleration does not change the response of the axis.

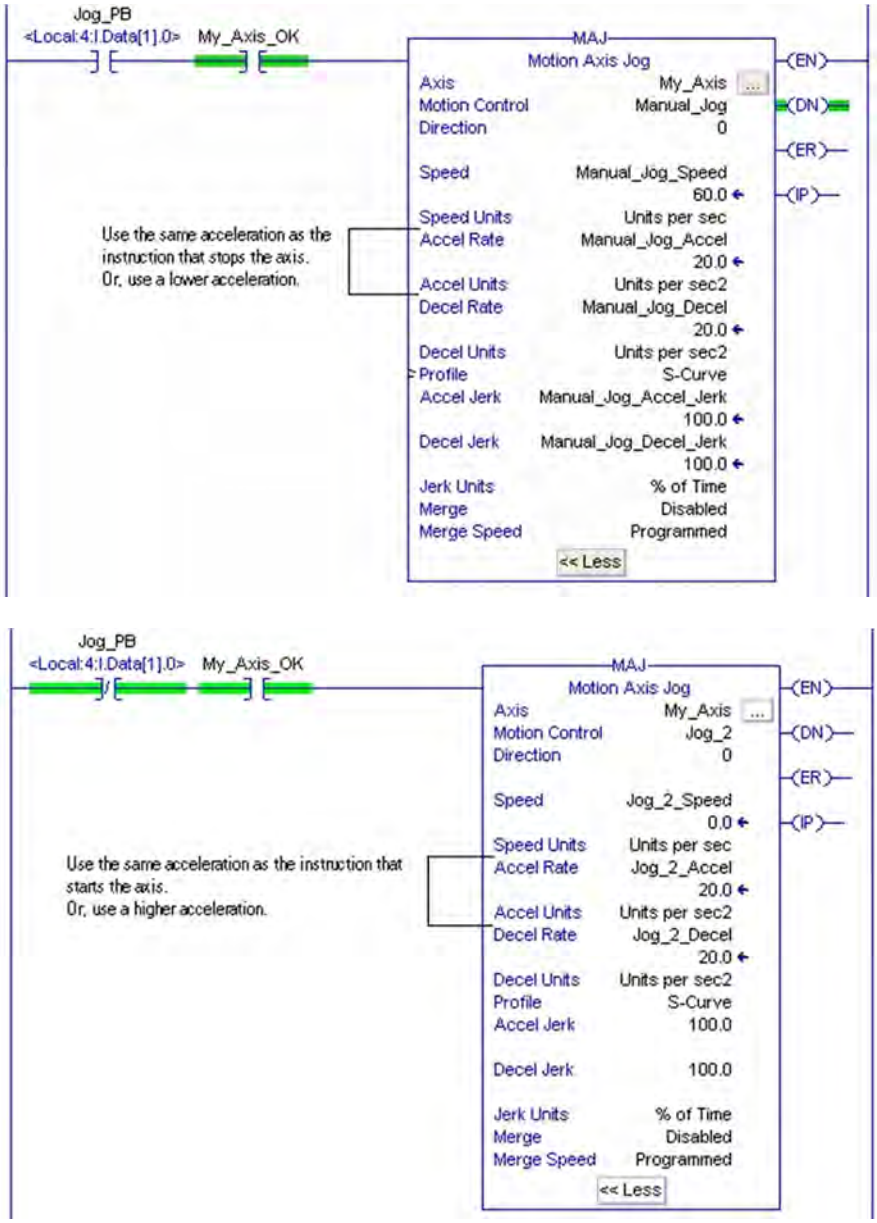
S-curve



The stopping instruction reduces the acceleration of the axis. It now takes longer to bring the acceleration rate to 0. The axis continues past its target speed until acceleration equals 0.

Corrective Action

Use a Motion Axis Stop (MAS) instruction to stop the axis or configure your instructions like the following:



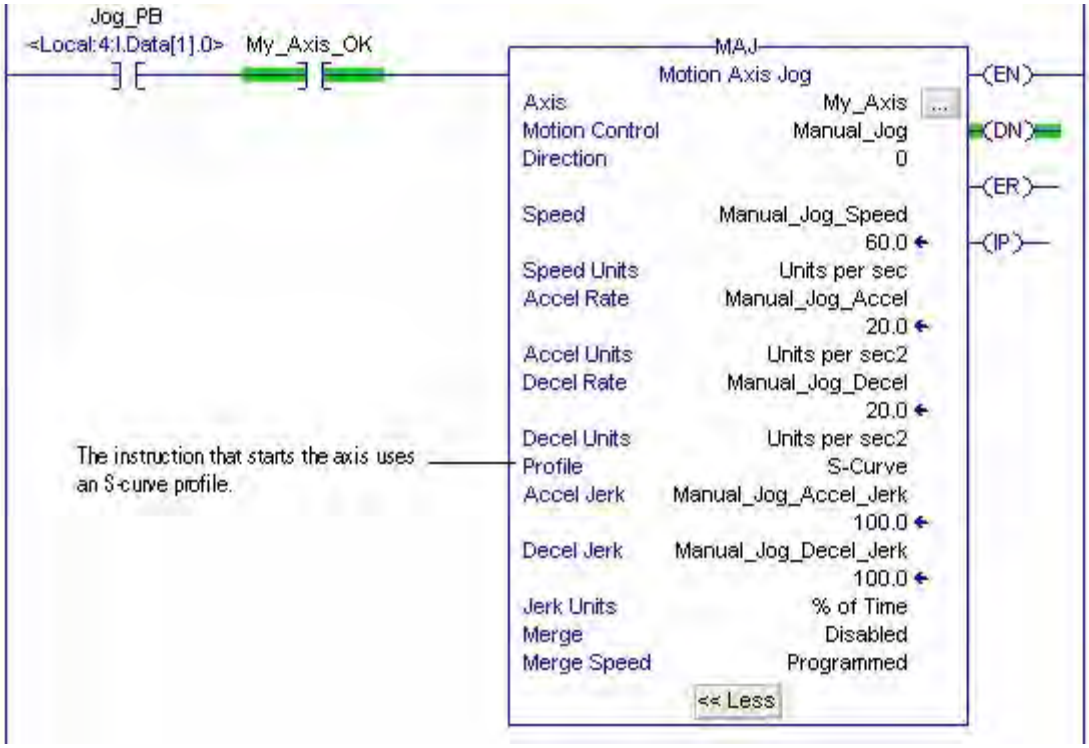
Why is there a delay when I stop and then restart a jog?

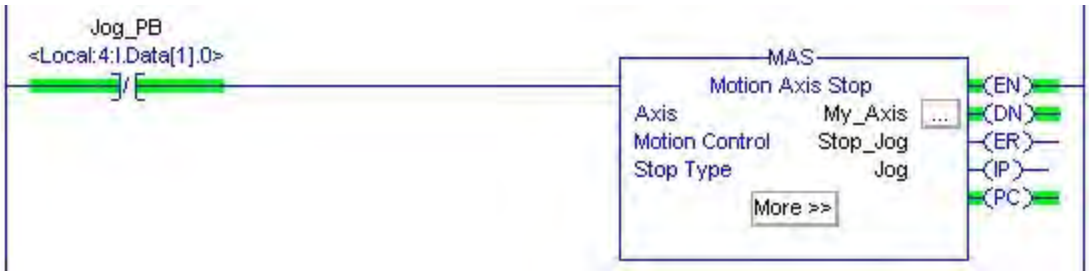
While an axis is jogging at its target speed, you stop the axis. Before the axis stops completely, you restart the jog. The axis continues to slow down before it speeds up.

Example

You use a Motion Axis Stop (MAS) instruction to stop a jog. While the axis is slowing down, you use a Motion Axis Jog (MAJ) instruction to start the axis again. The axis does not respond right away. It continues to slow down. Eventually it speeds back up to the target speed.

Look For





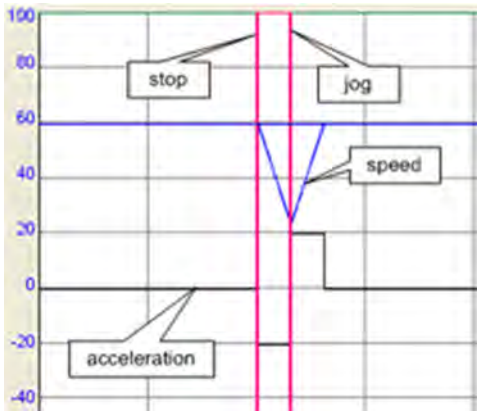
For Stop Type, the instruction that stops the axis keeps the S-curve profile. Suppose you use an Motion Axis Stop (MAS) instruction with the Stop Type set to Jog. In that case, the axis keeps the profile of the Motion Axis Jog (MAJ) instruction that started the axis.

Cause

When you use an S-curve profile, jerk determines the acceleration and deceleration time of the axis. An S-curve profile has to get acceleration to 0 before the axis can speed up again. The following trends show how the axis stops and starts with a trapezoidal profile and an S-curve profile.

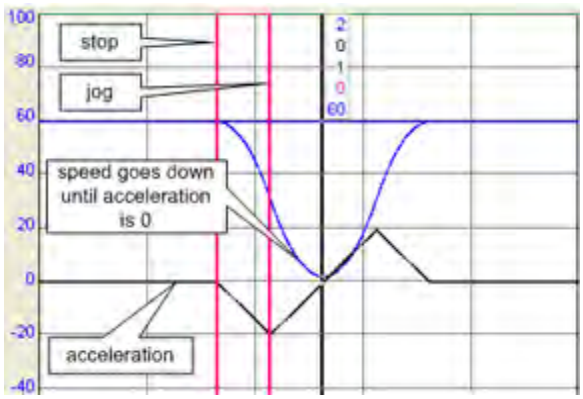
Start while decelerating

Trapezoidal



The axis speeds back up as soon as you start the jog again.

S-curve



The axis continues to slow down until the S-curve profile brings the acceleration rate to 0.

Corrective action

If your controller is this revision	Then	Result
15 or earlier	Increase the deceleration rate of the Motion Axis Jog (MAJ) instruction that starts the jog.	This increases the deceleration jerk. The axis stops the deceleration sooner at the higher deceleration jerk.
16 or later	Increase the deceleration jerk of the Motion Axis Jog (MAJ) instruction that starts the jog.	The axis stops the deceleration sooner at the higher deceleration jerk.

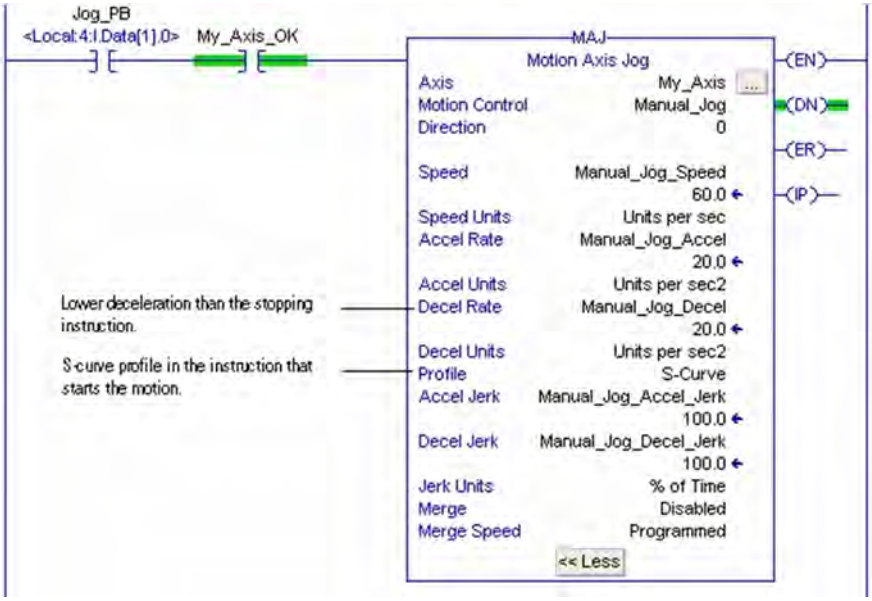
Why does my axis overshoot its position and reverse direction?

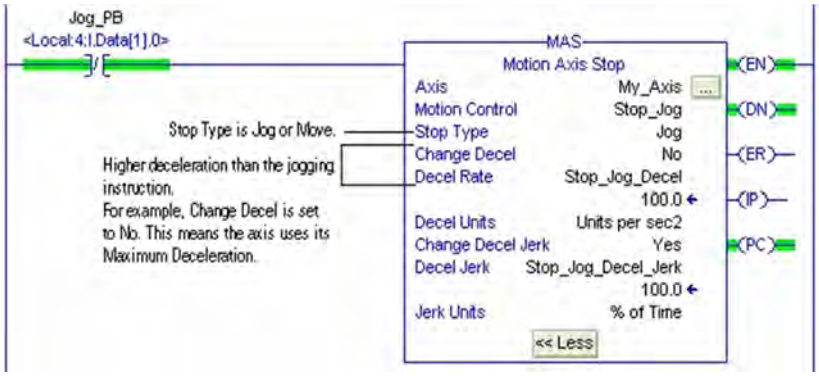
While an axis is moving to a target position, you change a parameter of the move. The axis overshoots its target position. Eventually the axis stops and moves back to its target position.

Example

You use a Motion Change Dynamics (MCD) instruction to reduce the deceleration while a Motion Axis Move (MAM) instruction is in process. The axis continues past the target position of the move, stops, and returns to the target position.

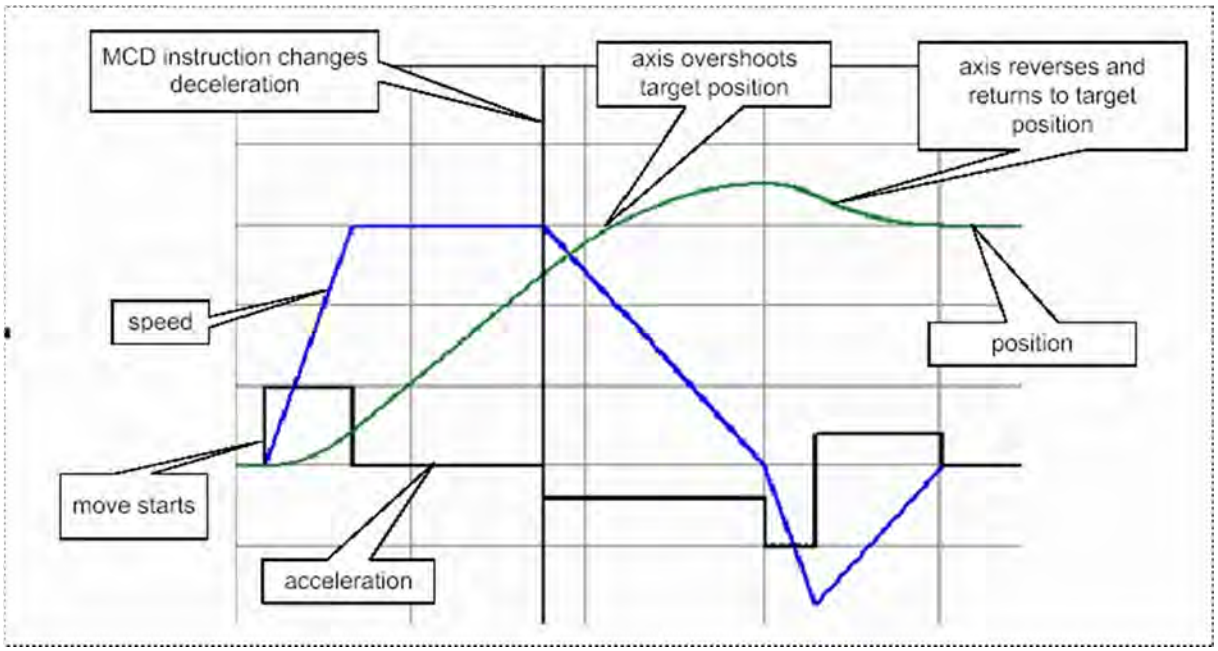
Look For





Cause

The axis does not have enough time at the new lower deceleration to stop at the target position. It stops past the target position. Then it corrects to get back to the target position.



Corrective action

To avoid overshooting position, do one of the following.

- Avoid decreasing the deceleration or deceleration jerk while an axis is decelerating along an S-curve profile.
- Avoid increasing the programmed speed while an axis is decelerating along an S-curve profile. This has the same effect as decreasing the deceleration jerk.

Test any changes in small increments to make sure a change does not cause an overshoot during normal operation.

Home an axis

Introduction for Home an Axis

Homing puts your equipment at a starting point for operation. This starting point is called the home position. Typically, you home your equipment when you reset it for operation. For more information about Homing, see Motion Axis Attributes on page 199.

Guidelines for homing

The following table provides descriptions of homing guidelines.

Guideline	Description
To move an axis to the home position, use Active homing.	Active homing turns on the servo loop and moves the axis to the home position. Active homing also: • Errors if there is any other motion on the axis. It does not stop other motion. • Uses a trapezoidal profile.
For a Feedback-only device, use Passive homing.	Passive homing does not move the axis. • Use passive homing to calibrate a Feedback-only axis to its marker. • If you use passive homing on a servo axis, turn on the servo loop and use a move instruction to move the axis.
If you have an absolute feedback device, consider Absolute homing.	If the motion axis hardware supports an absolute feedback device, Absolute Homing Mode may be used. The only valid Home Sequence for Absolute Homing Mode is Immediate. In this case, the absolute homing process establishes the true absolute position of the axis by applying the configured Home Position to the reported position of the absolute feedback device. Before execution of the absolute homing process via the MAH instruction, the axis must be in the Axis Ready state with the servo loop disabled.
For single-turn equipment, consider homing to a marker.	The marker homing sequence is useful for single-turn rotary and linear encoder applications because these applications have only one encoder marker for full axis travel.
For multi-turn equipment, home to a switch or switch and marker.	These homing sequences use a home limit switch to define the home position. • You need a home limit switch if the axis moves multiple revolutions when it runs. Otherwise the controller cannot tell which marker pulse to use. • For the most precise homing, use the switch and marker.

If your equipment cannot back up, use unidirectional homing.	With unidirectional homing, the axis does not reverse direction to move to the Home Position. To help insure that the Home operation is complete, consider using an offset.
	If these are not done, the axis position is still correct and accurate.
	• Use a Home Offset that is in the same direction as the Home Direction.
	• Use a Home Offset that is greater than the deceleration distance.
	• If the Home Offset is less than the deceleration distance:
	  • The axis simply slows to a stop. The axis does not reverse direction to move to the Home Position. In this case, the MAH instruction does not set the PC bit.
	  • On a rotary axis, the controller adds 1 or more revolutions to the move distance. This makes sure that the move to the Home Position is unidirectional.

Guideline	Description
Choose a starting direction for the homing sequence.	Which direction do you want to start the homing sequence in? • Positive direction: Choose a Forward direction. • Negative direction: Choose a Negative direction.

Active homing

When the axis **Homing Mode** is configured as **Active**, the physical axis is first activated for servo operation. The Home operation does not cancel other motion, but errors, Err=22.

You can home an axis using the configured Home **Sequence**, which may be **Immediate**, **Switch**, **Marker**, **Switch-Marker**, or **Torque-Level** homing. The Home Sequences result in the axis being jogged in the configured Home **Direction**. Using bidirectional homing, after the position is re-defined based on detection of the home event, the axis is automatically moved to the configured Home **Position**.

Important:	When unidirectional active homing is performed on a rotary axis and the Home **Offset** value is less than the deceleration distance when the home event is detected, it adds one or more revolutions to the move distance.

Passive homing

When the axis Homing **Mode** is configured as **Passive**, the MAH instruction re-defines the actual position of a physical axis on the next occurrence of the encoder marker, providing that Seq = Marker. Immediate, Switch, Switch-Marker and Torque Level homing is also allowed.

Passive homing is most commonly used to calibrate Feedback Only axes to their markers, but can also be used on Servo axes. Passive homing is identical to active homing to an encoder marker except that the Home command does not command any axis motion.

After initiating passive homing, the axis must be moved past the encoder marker for the homing sequence to complete properly. For closed-loop Servo axes, this may be accomplished with a MAM or MAJ instruction. For physical Feedback Only axes, motion cannot be commanded directly by the motion controller, and must be accomplished through other means.

Absolute homing

If the motion axis hardware supports an absolute feedback device, Absolute Homing Mode may be used. The only Home **Sequence** for an absolute Homing Mode is **immediate**. In this case, the absolute homing process establishes the true absolute position of the axis by applying the configured Home **Position**, to the reported position of the absolute feedback device. Before execution of the absolute homing process via the MAH instruction, the axis must be in the Axis Ready state with the servo loop disabled.

To successfully execute a MAH instruction on an axis configured for Active homing mode, the targeted axis must be configured as a Servo Axis Type. To successfully execute an MAH instruction, the targeted axis must be configured as a Servo or Feedback Only axis. If any of these conditions are not met, the instruction errors.

Important:	When the MAH instruction is initially executed, the In process (.IP) bit is set and the Process Complete (.PC) bit is cleared. The MAH instruction execution may take multiple scans to execute because it requires transmission of multiple messages to the motion module. Thus, the Done (.DN) bit is not set until after these messages are successfully transmitted. The .IP bit is cleared and the .PC bit is only set if the final axis position = the Home position.

This is a transitional instruction:

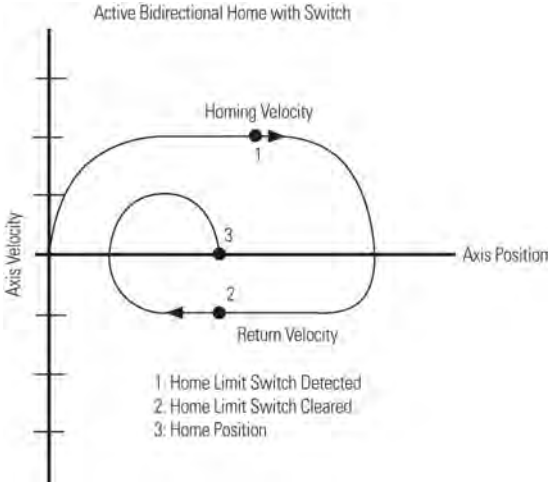
- In relay ladder, toggle the rung from cleared to set each time the instruction should execute.
- In structured text, condition the instruction so that it only executes on a transition.

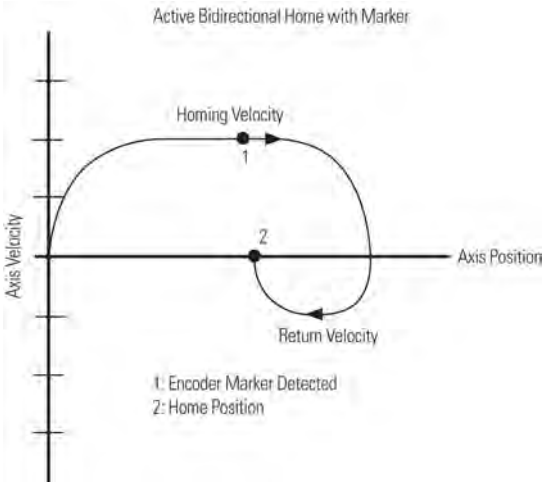
See Motion axis attributes on page 199 for more information.

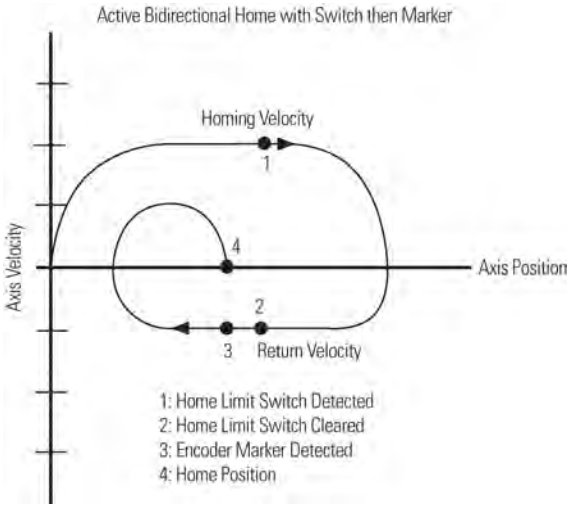
Examples

Active homing examples

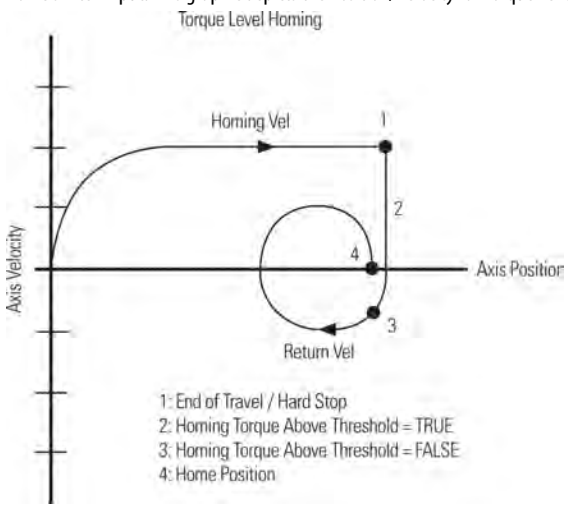
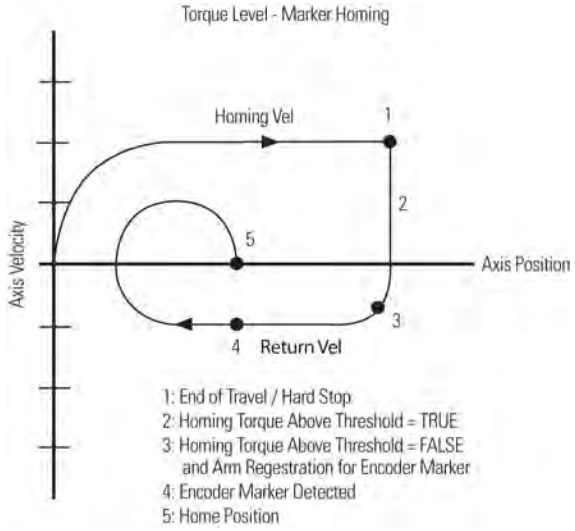
The following are Active Homing examples.

Sequence	Description
Active immediate home	This sequence sets the axis position to the Home Position without moving the axis. If the axis is not enabled, this sequence enables it. The feedback is working, therefore enabled.
Active home to switch in forward bidirectional	The switch homing sequence is useful for multi-turn rotary and linear applications. Ensure that the home switch is encountered in the direction of the home. If the axis is ON the home limit switch, it can be past the switch, but not on the switch.  The following steps occur during the sequence. 1. The axis moves in the Home Direction at the Home Speed to the home limit switch and decelerates to a stop (using the configured Maximum Deceleration Rate). It does not stop immediately. 2. If it is stopped as the Home Limit Switch, at position 2, when the sequence is started, it would be at the home position. The axis reverses direction and moves at the Home Return Speed until it clears the home limit switch and then stops. 3. The axis moves back to the home limit switch or it moves to the Offset position. The axis moves at the Home Return Speed. If the axis is a Rotary Axis, it moves on the shortest path to the Home Position (that is, no more than ½ revolution). If the axis is past the home limit switch at the start of the homing sequence, the axis reverses direction and starts the return leg of the homing sequence. Again, it decelerates to a stop. Use a Home Return Speed that is slower than the Home Speed to increase the homing accuracy. The accuracy of this sequence depends on the return speed and the delay to detect the transition of the home limit switch. Uncertainty = Home Return Speed x delay to detect the home limit switch. **Example**: Suppose your Home Return Speed is 0.1 in./s and it takes 10 ms to detect the home limit switch. Uncertainty = 0.1 in./s x 0.01 s = 0.001 in. The mechanical uncertainty of the home limit switch also affects the homing accuracy.

Sequence	Description
Active home to marker in forward bidirectional	The marker homing sequence is useful for single-turn rotary and linear encoder applications because these applications have only one encoder marker for full axis travel.  The following steps occur during the sequence. 1. The axis moves in the Home Direction at the Home Speed to the marker and decelerates to a stop. 2. The axis moves back to the marker or it moves to the Offset position. The axis moves at the Home Return Speed. If the axis is a Rotary Axis, it moves along the shortest path to the Home Position (that is, no more than ½ revolution). The accuracy of this homing sequence depends on the homing speed and the delay to detect the marker transition. Uncertainty = Home Speed x delay to detect the marker. **Example**: Suppose your Home Speed is 1 in./s and it takes 1 µs to detect the marker. Uncertainty = 1 In./s x 0.000001 s = 0.000001 in.

Sequence	Description
Active home to switch and marker in forward bidirectional	This is the most precise active homing sequence available.  The following steps occur during the sequence. 1. The axis moves in the Home Direction at the Home Speed to the home limit switch and decelerates to a stop. 2. The axis reverses direction and moves at the Home Return Speed until it clears the home limit switch. 3. The axis keeps moving at the Home Return Speed until it gets to the marker. 4. The axis moves back to the marker or it moves to the Offset position. The axis moves at the Home Return Speed. If the axis is a Rotary Axis, it moves along the shortest path to the Home Position (that is, no more than ½ revolution). If the axis is ON the home limit switch at the start of the homing sequence, the axis reverses direction and starts the return leg of the homing sequence.
Active home to switch in forward unidirectional	This active homing sequence is useful when an encoder marker is not available and unidirectional motion is required or proximity switch is being used. The following steps occur during the sequence. 1. The axis moves in the Home Direction at the Home Speed to the home limit switch. 2. A decel position is calculated using the Home Offset and the decel distance. The axis keeps moving to the decel position and then decelerates to a stop. 3. The axis moves to the Home Offset position if it is in the same direction as the Home Direction. • When your position mode is linear, it decelerates to a stop. You may not be at your home position but you are correctly referenced to your home position. • When your position mode is Rotary, the rotary turns as many times that it needs to decelerate and finish at the home position.
Active home to marker in forward unidirectional	This active homing sequence is useful for single-turn rotary and linear encoder applications when unidirectional motion is required. The following steps occur during the sequence. 1. The axis moves in the Home Direction at the Home Speed to the marker. 2. If the axis is linear, it decelerates to a stop, unless the home offset is greater than the distance required to decelerate; then the home offset is applied. If the axis is rotary, it adds as many revolutions as necessary so it decelerates and stops at the home position. 3. The axis moves to the Home Offset position if it is in the same direction as the Home Direction.

Sequence	Description
Active home to switch and marker in forward unidirectional	This active homing sequence is useful for multi-turn rotary applications when unidirectional motion is required. The following steps occur during the sequence. 1. The axis moves in the Home Direction at the Home Speed to the home limit switch. 2. The axis keeps moving at the Home Speed until it gets to the marker. 3. If the axis is linear, it decelerates to a stop, unless the home offset is greater than the distance required to decelerate; then the home offset is applied. If the axis is rotary, it adds as many revolutions as necessary so it decelerates and stops at the home position.

Sequence	Description
Active Home to Torque	The Home to Torque Level sequence is a type of homing used when a hard stop is going to be used as the home position, as in a linear actuator. The occurrence of the hard stop is detected by the drive when the output torque to the motor reaches or exceeds the torque level specified by the user. Since the home to torque level sequence relies on the mechanical end of travel for operation, Unidirectional homing is not possible. Only Forward Bidirectional and Reverse Bidirectional are allowed. In Torque Level homing, the torque event is the trigger. The motion planner decelerates the axis to a stop and reverses direction. The torque event is usually some type of hard stop. Because of this, the physical axis cannot move, but the position command is changing. This causes the Position error to increase. If the distance required to decelerate is greater than the Position error Tolerance, an Excessive Position error exception can occur, possibly canceling the home operation. A delay filter is implemented in the drive to reduce any false/nuisance triggers when there is a spike in the torque feedback upon enabling or jogging the motor under load. Torque Level homing is very similar to Home Switch homing, with the exception that the torque level is used instead of the home switch input. This graphic depicts the Position/Velocity for Torque Level Homing.  Torque Level-Marker homing is very similar to Home Switch-Marker homing, with the exception that the torque level is used instead of the home switch input. This graphic depicts the Position/Velocity for Torque Level-Marker Homing. 

Passive homing examples

The following table provides passive homing examples.

Sequence	Description
Passive Immediate Home	This is the simplest passive homing sequence type. When this sequence is performed, the controller immediately assigns the Home Position to the current axis actual position. This homing sequence produces no axis motion.
Passive Home with Switch	This passive homing sequence is useful when an encoder marker is not available or a proximity switch is being used. When this sequence is performed in the Passive Homing Mode, an external agent moves the axis until the home switch is detected. The position is preset to the Home position plus Offset Value at the moment when the switch is hit. The Offset value should be set to 0 if no Home Offset offset is wanted.
Passive Home with Marker	This passive homing sequence is useful for single-turn rotary and linear encoder applications. When this sequence is performed in the Passive Homing Mode, an external agent moves the axis until the marker is detected. The position is preset to the Home position plus Offset Value at the moment when the switch is hit. The Offset value should be set to 0 if no Home Offset offset is wanted.
Passive Home with Switch then Marker	This passive homing sequence is useful for multi-turn rotary applications. When this sequence is performed in the Passive Homing Mode, an external agent moves the axis until the home switch and then the first encoder marker is detected. The position is preset to the Home position plus Offset Value at the moment when the switch is hit. The Offset value should be set to 0 if no Home Offset offset is wanted.

Homed Status

The Homed Status bit is set by the MAH instruction upon successful completion of the configured homing sequence. This bit indicates that an absolute machine reference position has been established. When this bit is set, operations that require a machine reference, such as Software Overtravel checking, can be meaningfully enabled.

For CIP Drive axis data types, the Homed Status bit is cleared under the following condition.

- MRP instruction

For non-CIP Drive axis data types, the Homed Status bit is cleared under the following conditions.

- Download
- Control power cycle

- Re-connection to Motion Module
- Feedback Loss Fault
- Shutdown

Feedback Integrity

This bit, when set, indicates that the feedback device is accurately reflecting axis position. The bit is set at power-up assuming that the feedback device passes any power-up self test required. If during operation, a feedback exception occurs that could impact the fidelity of axis position, the bit is immediately cleared. The bit remains clear until a fault reset is executed by the drive or the drive is power cycled. Note that the Feedback Integrity bit behavior applies to absolute and incremental feedback device operation.

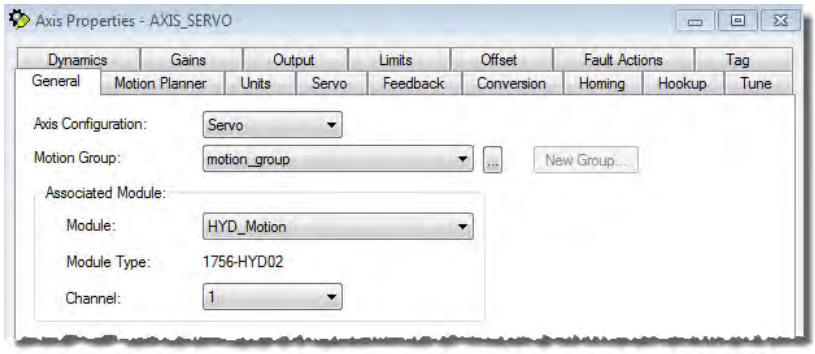
Axis properties

Introduction for Axis Properties

Use this appendix for a description of the properties of an axis. For a description of the Axis_CIP_Drive properties, see Integrated Motion on the Ethernet/IP Network Configuration and Startup, publication MOTION-UM003.

General tab – AXIS_SERVO

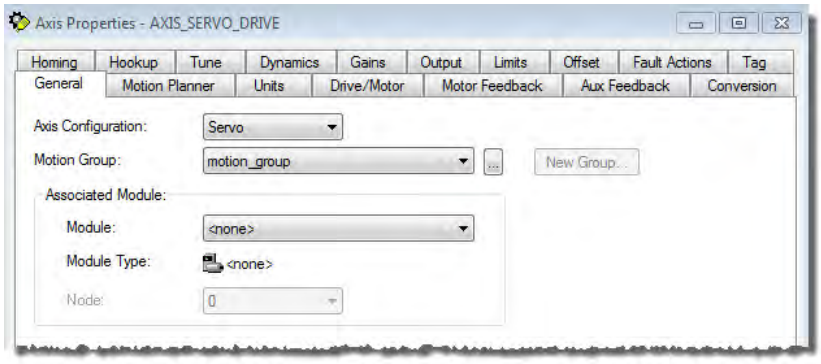
The following **General** dialog box below is for an AXIS_SERVO data type.

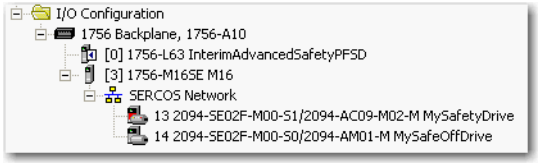


Item	Description	
Axis Configuration	Selects and displays the intended use of the axis.	
	Feedback Only	If the axis is to be used only to display position information from the feedback interface. This selection minimizes the display of axis properties tabs and parameters. The tabs **Servo**, **Tune**, **Dynamics**, **Gains**, **Output**, **Limits**, and **Offset** are not displayed.
	Servo	If the axis is to be used for full servo operation. This selection maximizes the display of axis properties tabs and parameters.
Motion Group	Selects and displays the Motion Group to which the axis is associated. An axis assigned to a Motion Group appears in the Motion Groups branch of the **Controller Organizer**, under the selected Motion Group sub-branch. Selecting <none> terminates the Motion Group association, and moves the axis to the Ungrouped Axes sub-branch of the Motion Groups branch.	
The Associated Module selection (selected on the **General** tab) determines available catalog numbers.		
Module	Selects and displays the name of the motion module to which the axis is associated. Displays <none> if the axis is not associated with any motion module.	
Module Type	Displays a module icon and the name of the SERCOS drive to which the axis is associated. Displays <none> if the axis is not associated with any drive. If the associated drive is a Kinetix Safety drive, a portion of the module icon is red to signify its safety significance.	
Channel	Selects and displays the 1756-M02AE motion module channel, 0 or 1, to which the axis is assigned. Disabled when the axis is not associated with any motion module.	

General tab - AXIS_SERVO_DRIVE

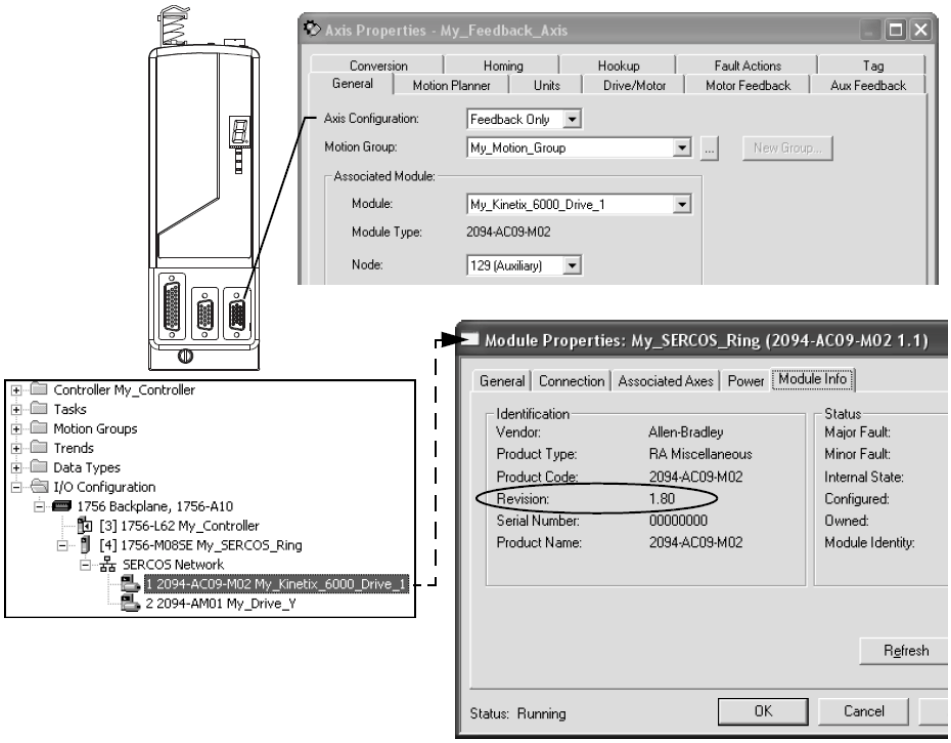
The following image is an example of the **General** tab for an AXIS_SERVO DRIVE Data Type.



Item	Description	
Axis Configuration	Selects and displays the intended use of the axis.	
	Feedback Only	If the axis is to be used only to display position information from the feedback interface. This selection minimizes the display of axis properties tabs and parameters. The tabs **Tune**, **Dynamics**, **Gains**, **Output**, **Limits**, and **Offset** are not displayed.
	Servo	If the axis is to be used for full servo operation. This selection maximizes the display of axis properties tabs and parameters.
Motion Group	Selects and displays the Motion Group to which the axis is associated. An axis assigned to a Motion Group appears in the Motion Groups branch of the **Controller Organizer**, under the selected Motion Group sub-branch. Selecting <none> terminates the Motion Group association, and moves the axis to the Ungrouped Axes sub-branch of the Motion Groups branch.	
Module	Selects and displays the name of the SERCOS drive to which the axis is associated. Displays <none> if the axis is not associated with any drive.	
Module Type	Displays a module icon and the name of the SERCOS drive to which the axis is associated. Displays <none> if the axis is not associated with any drive. If the associated drive is a Kinetix Safety drive, a portion of the module icon is red to signify its safety significance. Following is a sample of the module icon (from the I/O configuration folder in Logix Designer application) for: • Kinetix 6000 Advanced Safety Drive (S1) • Kinetix 6000 Enhanced Safe Torque-Off Drive (S0) 	
Node	Displays the base node of the associated SERCOS drive. This is unavailable when the axis is not associated with any drive.	

Node with a Kinetix 6000 drive

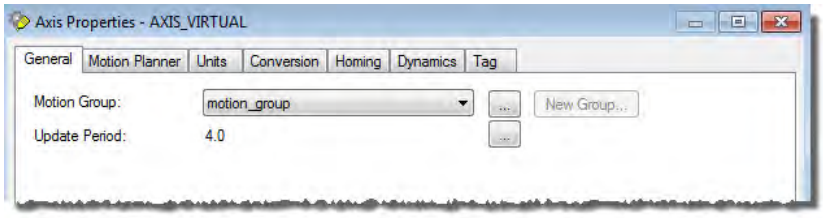
If you use the auxiliary feedback port of a Kinetix 6000 drive as a feedback-only axis, the drive must have firmware revision 1.80 or later.



When a Kinetix 6000 drive is designated in the **Associated Module** box, there is an additional option for the **Node** value. It is the node associated with the drive plus 128 with (Auxiliary) after the number. The range is **129** to **234**. When the Auxiliary Node assignment is chosen, the axis configuration is changed to **Feedback Only** on the **General** tab and an asterisk (*) appears next to General. This also places an asterisk (*) on the **Aux Feedback** tab and you must click that tab and choose values. On the **Drive/Motor** tab the **Loop Configuration** is changed to **Aux Feedback Only**.

General tab - AXIS_VIRTUAL

The following image is an example of the **General** tab for an AXIS_VIRTUAL Data Type.



Motion Group

You can select the Motion Group to which you want to associate the axis. An axis assigned to a Motion Group appears in the Motion Groups branch of the **Controller Organizer**. When you select 'none', it terminates the Motion Group association, and moves the axis to the Ungrouped Axes in the **Controller Organizer**.

MOTION_GROUP structure

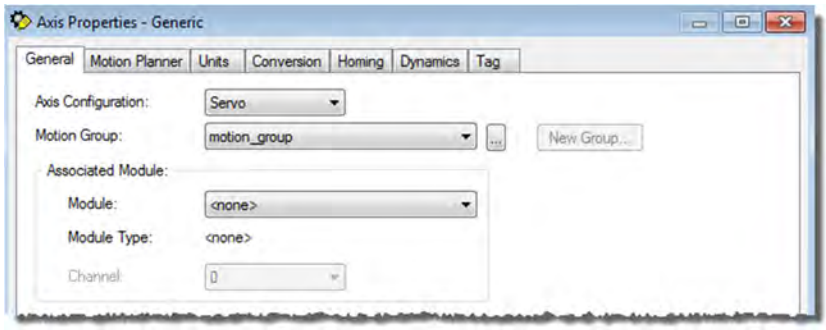
There is one MOTION_GROUP data type per controller. This structure contains status and configuration information about the motion group.

Enumerations	Data Type	Description			
GroupStatus	DINT	The status bits for the group.			
		Bit	**Number**	**Data Type**	**Description**
		InhibitStatus	00	DINT	Inhibit status.
		GroupSynced	01	DINT	Synchronization status.
		AxisInhibitStatus	02	DINT	
		-no-tag	02	DINT	Timer Event started.
		Reserved	03…31		
MotionFault	DINT	The motion fault bits for the group.			
		Bit	**Number**	**Data Type**	**Description**
		ACAsyncConnFault	00	DINT	Asynchronous connection fault.
		ACSyncConnFault	01	DINT	Synchronous connection fault.
		Reserved	02…31		
ServoFault	DINT	The servo-module fault bits for the group.			
		Bit	**Number**	**Data Type**	**Description**
		POtrvlFault	00	DINT	Positive overtravel fault.
		NOtrvlFault	01	DINT	Negative overtravel fault.
		PosErrorFault	02	DINT	Position error fault.

Enumerations	Data Type	Description			
		EncCHALossFault	03	DINT	Encoder channel A loss fault.
		EncCHBLossFault	04	DINT	Encoder channel B loss fault.
		EncCHZLossFault	05	DINT	Encoder channel Z loss fault.
		EncNsFault	06	DINT	Encoder noise fault.
		DriveFault	07	DINT	Drive fault.
		Reserved	08…31		
		Bit	**Number**	**Data Type**	**Description**
		SyncConnFault	00	DINT	Synchronous connection fault.
		HardFault	01	DINT	Servo hardware fault.
		Reserved	02…31		
GroupFault	DINT	The fault bits for the group.			
		Bit	**Number**	**Data Type**	**Description**
		GroupOverlapFault	00	DINT	Group task overlap fault.
		CSTLossFault	01	DINT	The controller has lost synchronization with the CST master.
		GroupTaskLoadingFault	02	DINT	The group base update period is too low, user application tasks are not getting enough time to execute.
		Reserved	03…31		
AxisFault	DINT	The fault bits for the axis.			
		Bit	**Number**	**Data Type**	**Description**
		PhysicalAxisFault	00	DINT	A Servo or Drive fault has occurred.
		ModuleFault	01	DINT	A serious fault has occurred with the motion module associated with the selected axis. Usually affects all axes associated with the motion module.
		ConfigFault	02	DINT	One or more axis attributes associated with a motion module or drive has not been successfully updated to match the value of the corresponding attribute of the local controller.
		Reserved	03…31		

General tab – AXIS_GENERIC

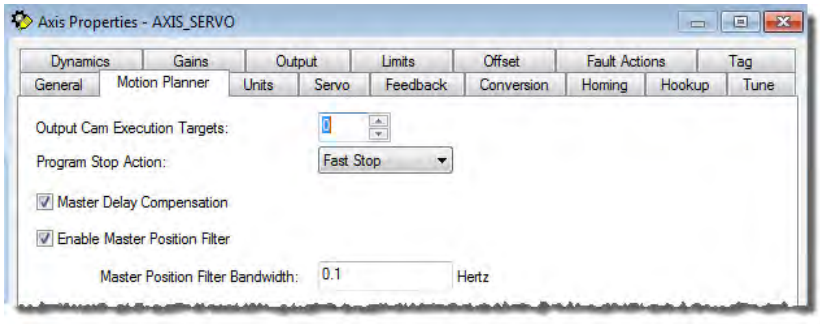
The following image is an example of the **General** tab for an AXIS_GENERIC data type.



Item	Description	
Axis Configuration	Selects and displays the intended use of the axis.	
	Feedback Only	If the axis is to be used only to display position information from the feedback interface. This selection minimizes the display of axis properties tabs and parameters. The **Dynamics** tab is not available.
	Servo	If the axis is to be used for full servo operation. This selection maximizes the display of axis properties tabs and parameters.
Motion Group	Selects and displays the Motion Group to which the axis is associated. An axis assigned to a Motion Group appears in the Motion Groups branch of the **Controller Organizer**, under the selected Motion Group sub-branch. Selecting <none> terminates the Motion Group association, and moves the axis to the Ungrouped Axes sub-branch of the Motion Groups branch.	
Module	Selects and displays the name of the motion module to which the axis is associated. Displays <none> if the axis is not associated with any motion module.	
Channel	Selects and displays the motion module channel, 0 or 1, to which the axis is assigned. Disabled when the axis is not associated with any motion module.	

Motion Planner tab

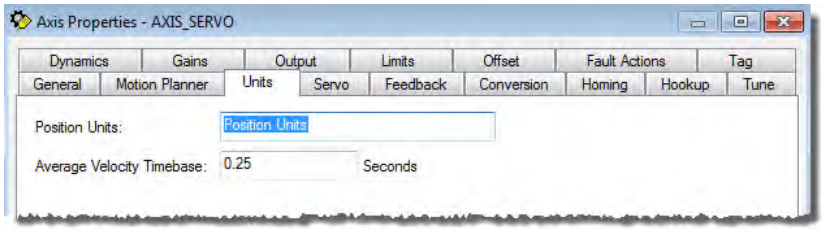
The **Motion Planner** tab is where you set and edit the number of **Output Cam Execution Targets** and the **Program Stop Action**, select and clear the **Master Delay Compensation** and **Enable Master Position Filter**, and set the bandwidth for **Master Position Filter Bandwidth**. The **Motion Planner** tab has the same fields regardless of the type of axis.



Item	Description	
Output Cam Execution Targets	Determines how many Output Cam execution nodes (instances) are created for an axis. The **Execution Target** parameter for the MAOC/MDOC instructions specifies which of the configured execution nodes the instruction affects. In addition, the number specified in the **Axis Properties** dialog box specifies the number of instances of Output Cam in which the value of zero means 'none'. The value specified for Execution Target in the MAOC instruction references an instance in which a value of zero selects the first instance.	
Program Stop Action	Selects how an axis is stopped when the processor undergoes a mode change, or when an explicit Motion Group Programmed Stop (MGPS) instruction is executed. You can apply Program Stop Action when an MSG is programmed to Stop type.	
	Fast Disable	The axis is decelerated to a stop using the current configured value for maximum deceleration. Servo action is maintained until the axis motion has stopped, at which time the axis is turned off (that is, **Drive Enable** is not checked, and **Servo Action** is not checked).
	Fast Shutdown	The axis is decelerated to a stop using the current configured value for maximum deceleration. Once the axis motion is stopped, the axis is placed in the shutdown state (that is, Drive Enable is not checked, Servo Action is not checked, and the OK contact is opened). To recover from this state, a Shutdown reset instruction must be executed.
	Fast Stop	The axis is decelerated to a stop using the current configured value for maximum deceleration. Servo action is maintained after the axis motion has stopped. This mode is useful for gravity or loaded systems, where servo control is needed at all times.
	Hard Disable	The axis is immediately disabled (that is, Drive Enable is disabled, Servo Action is disabled, but the OK contact is left closed). Unless the drive is configured to provide some form of dynamic breaking, this results in the axis coasting to a stop.
	Hard Shutdown	The axis is immediately placed in the shutdown state. Unless the drive is configured to provide some form of dynamic breaking, this results in the axis coasting to a stop. To recover from this state, a Shutdown reset instruction must be executed.
Master Delay Compensation	Enables or disables Master Delay Compensation. The default setting is Disabled. It must be selected to enable master delay compensation. Master Delay Compensation balances the delay time between reading the master axis command position and applying the associated slave command position to the slave's servo loop. It ensures that the slave axis command position accurately tracks the actual position of the master axis; that is, zero tracking error. If the axis is configured for Feedback only, Master Delay Compensation should be disabled.	
Enable Master Position Filter	Enables or disables the Master Position Filter. The default is disabled. It must be selected to enable position filtering. Master Position Filter effectively filters the specified master axis position input to the slave axis's gearing or position camming operation. The filter smooths out the actual position signal from the master axis, and thus smooths out the corresponding motion of the slave axis. When this feature is enabled the **Master Position Filter Bandwidth** field is enabled.	
Master Position Filter Bandwidth	Enabled when the **Enable Master Position Filter** check box is selected. This field controls the bandwidth for master position filtering. Enter a value in Hz to set the bandwidth for the Master Position Filter. **Important:** A value of zero for Master Position Filter Bandwidth effectively disables the master position filtering.	

Units tab

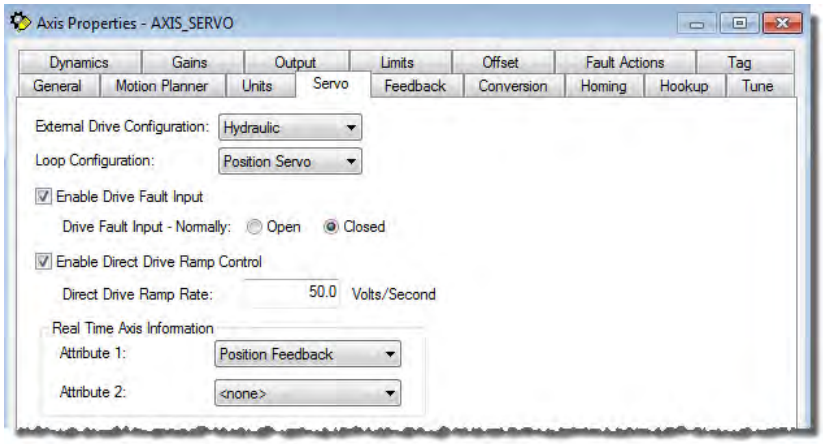
The **Units** tab is the same for all axis data types. Use this tab to determine the units to define your motion axis.



Item	Description
Position Units	User-defined engineering units (rather than feedback counts) used for labeling all motion-related values, for example, position and velocity. These position units can differ for each axis. Position units should be chosen for maximum ease of use in your application. For example, linear axes might use position units of Inches, Meters, or mm whereas rotary axes might use units of Revs or Degrees.
Average Velocity Timebase	Specifies the time (in seconds) to be used for calculating the average velocity of the axis. This value is computed by taking the total distance the axis travels in the amount of time specified and dividing this value by the timebase. The average velocity timebase value should be large enough to filter out the small changes in velocity that would result in a noisy velocity value, but small enough to track significant changes in axis velocity. A value of 0.25 to 0.50 seconds should work well for most applications.

Servo tab - AXIS_SERVO

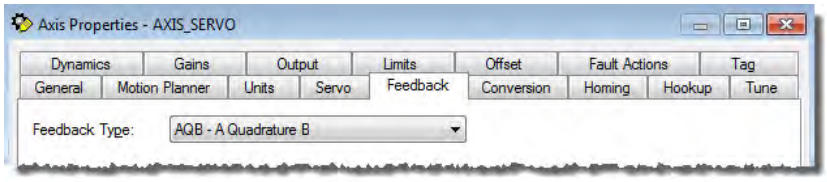
The following image is an example of the **Servo** tab for AXIS_SERVO.



Item	Description
External Drive Configuration	Select the drive type for the servo loop: • **Velocity** - Disables the servo module's internal digital velocity loop. • **Torque** - The servo module's internal digital velocity loop is active, which is the required configuration for interfacing the servo axis to a torque loop drive. • **Hydraulic** - Enables hydraulic servo application features.
Loop Configuration	Select the configuration of the servo loop. For this release, only **Position Servo** is available.
Enable Drive Fault Input	Select this check box if you wish to enable the Drive Fault Input. When active the motion module receives notice whenever the external drive detects a fault.
Drive Fault Input	Specifies the usual state of the drive fault input when a fault is detected on the drive. • **Normally Open** – When a drive fault is detected, it opens its drive fault output contacts. • **Normally Closed** – When a drive fault is detected, it closes its drive fault output contacts.
Enable Direct Drive Ramp Control	Select this check box to set the Direct Drive Ramp Rate in volts per second for when an MDO instruction is executed.
Direct Drive Ramp Rate	The Direct Drive Ramp Rate is a slew rate for changing the output voltage when a Direct Drive On (MDO) instruction is executed. A Direct Drive Ramp Rate of 0 disables the output rate limiter, letting the Direct Drive On voltage be applied directly.
Attribute 1/Attribute 2	Select up to two axis attributes whose statuses are transmitted with, for example, the actual position data to the Logix processor. The values of the selected attributes can be accessed through the standard GSV or Get Attribute List service. They can also be accessed using template data. The servo status data is updated each base update period. If a GSV is done to one of these servo status attributes without having selected this attribute through the Drive Info Select attribute, the attribute value is static and does not reflect the true value in the servo module.

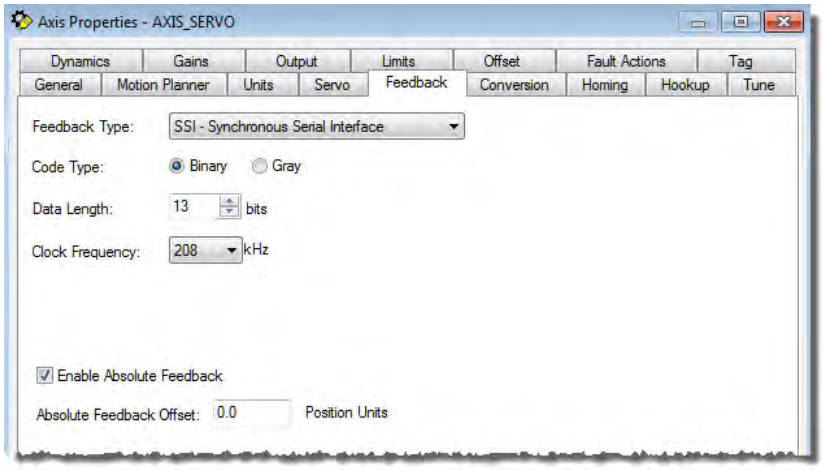
Feedback Tab – AXIS_SERVO

Use the **Feedback** tab to select the type of Feedback used with your Servo axis.



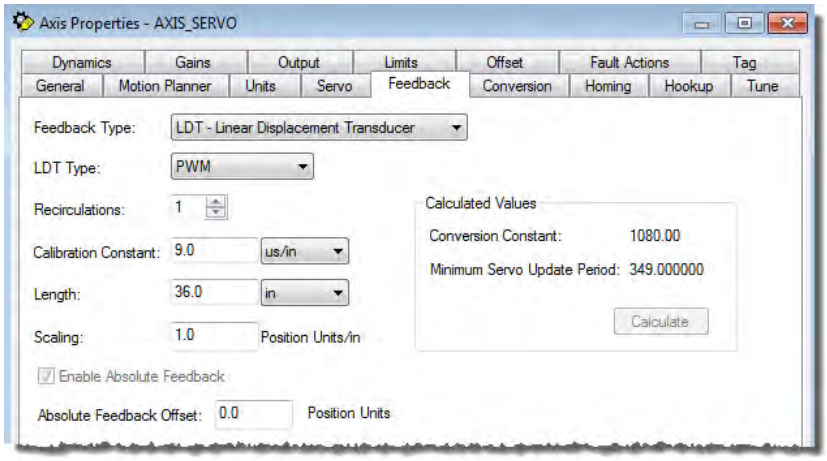
Item	Description	
Feedback Type	Select the Feedback type for your current configuration. Your options depend on the motion module to which the axis is associated.	
	A Quadrature B Encoder Interface (AQB)	The 1756-M02AE servo module provides interface hardware to support incremental quadrature encoders equipped with standard 5-Volt differential encoder-interface signals. The AQB option has no associated attributes to configure.

Item	Description	
	Synchronous Serial Interface (SSI)	The 1756-M02AS servo module provides an interface to transducers with Synchronous Serial Interface (SSI) outputs. SSI outputs use standard 5V differential signals (RS422) to transmit information from the transducer to the controller. The signals consist of a Clock generated by the controller and Data generated by the transducer.
	Linear Displacement Transducer (LDT)	The 1756-HYD02 Servo module provides an interface to the Linear Magnetostrictive Displacement Transducer, or LDT. A Field Programmable Gate Array (FPGA) is used to implement a multi-channel LDT Interface. Each channel is functionally equivalent and is capable of interfacing to an LDT device with a maximum count of 240,000. The LDT interface has transducer failure detection and digital filtering to reduce electrical noise.



Item	Description
Feedback Type	Feedback Type is set to **SSI - Synchronous Serial Interface**.
Code Type	The type of code, **Binary** or **Gray**, used to report SSI output. If the module's setting does not match the feedback device, the positions jump around erratically as the axis moves.
Data Length	The length of output data in a specified number of bits between 8 and 31. The data length for the selected feedback device can be found in its specifications.
Clock Frequency	Sets the clock frequency of the SSI device to 208 (default) or 625 kHz. When the higher clock frequency is used, the data from the feedback device is more recent, but the length of the cable to the transducer must be shorter than with the lower frequency.
Enable Absolute Feedback	The default is enabled (checked). If Enable Absolute Feedback is set, the servo module adds the **Absolute Feedback Offset** to the current position of the feedback device to establish the absolute machine reference position. Absolute feedback devices retain their position reference even through a power-cycle; therefore, the machine reference system can be restored at powerup.

Item	Description
Absolute Feedback Offset	If Absolute feedback is enabled, this field becomes active. You can enter the amount of offset, in position units, to be added to the current position of the Feedback device. The SSI is an absolute feedback device. To establish a value for the Offset, the MAH instruction can be executed with the Home **Mode** set to Absolute. When executed, the module computes the Absolute Feedback Offset as the difference between the configured value for Home Position and the current absolute feedback position of the axis. The computed Absolute Feedback Offset is immediately applied to the axis upon completion of the MAH instruction. The actual position of the axis is re-referenced during execution of the MAH instruction; therefore, the servo loop must not be active. If the servo loop is active, the MAH instruction generates an error. When the **Enable Absolute Feedback** is disabled, the servo module ignores the Absolute Feedback Offset and treats the feedback device as an incremental position transducer. A homing or redefine position operation is required to establish the absolute machine reference position. The Absolute Home Mode is invalid. If using single-turn or multi-turn Absolute SSI Feedback transducers, see the **Homing** tab information for important details concerning Absolute feedback transducer's marker reference.

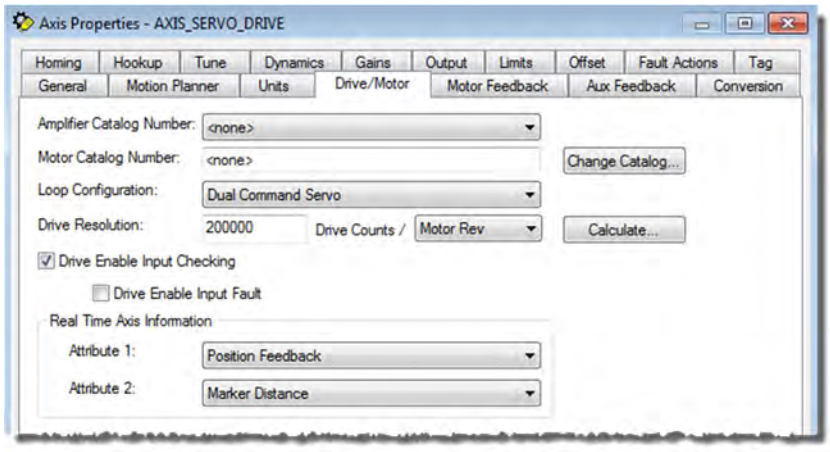


Item	Description
Feedback Type	Feedback Type is set to **LDT - Linear Displacement Transducer**.
LDT Type	This field selects the type of LDT to use to provide feedback to the Hydraulic module. The available types are PWM, Start/Stop Rising, or Start/Stop Falling.
Recirculations	Use this field to set the number of repetitions to use to acquire a measurement from an LDT.
Calibration Constant	This is a number that is engraved on the LDT by the manufacturer. It specifies the characteristics of the individual LDT. Each LDT has its own calibration constant; therefore, if you change the LDT, you must change the Calibration constant.
Length	Defines the stroke of travel of the hydraulic cylinder. The length value is used with the number of recirculations to determine the minimum servo update period.
Scaling	Scaling defines the relationship between the LDT unit of measure (length field) and the unit of measure defined at the **Units** tab.
Enable Absolute Feedback	This field is grayed out because it is active when Feedback Type is LDT.

Item	Description	
Absolute Feedback Offset	Enter the amount of offset, in position units, to be added to the current position of the LDT. The LDT is an absolute feedback device. To establish a value for the Offset, the MAH instruction can be executed with the Home Mode set to Absolute. When executed, the module computes the Absolute Feedback Offset as the difference between the configured value for Home Position and the current absolute feedback position of the axis. The computed Absolute Feedback Offset is immediately applied to the axis upon completion of the MAH instruction. The actual position of the axis is re-referenced during execution of the MAH instruction; therefore, the servo loop must not be active. If the servo loop is active, the MAH instruction generates an error. When the **Enable Absolute Feedback** is disabled, the servo module ignores the **Absolute Feedback Offset** and treats the feedback device as an incremental position transducer. A homing or redefine position operation is required to establish the absolute machine reference position. The Absolute Home Mode is invalid.	
Calculated Values	Conversion Constant	The **Conversion Constant** is calculated from the values entered on the **Feedback** screen when you click **Calculate**. This calculated value must be typed into the **Conversion Constant** field on the **Conversion** tab as it is not automatically updated.
	Minimum Servo Update Period	The **Minimum Servo Update period** is calculated based on the values entered for **Recirculations** and **Length** on the **Feedback** tab. When these values are changed, clicking **Calculate** recalculates the **Minimum Servo Update Period** based on the new values.
	Calculate Button	Calculate becomes active whenever you make changes to the values on the Feedback tab. Clicking **Calculate** recalculates the **Conversion Constant** and **Minimum Servo Update Period** values. However, you must then re-enter the **Conversion Constant** value at the **Conversion** tab as the values are not updated automatically.

Drive/Motor tab - AXIS_SERVO_DRIVE

Use the **Drive/Motor** tab to configure the servo loop for an AXIS_SERVO_DRIVE axis, and open the **Change Catalog** dialog box.

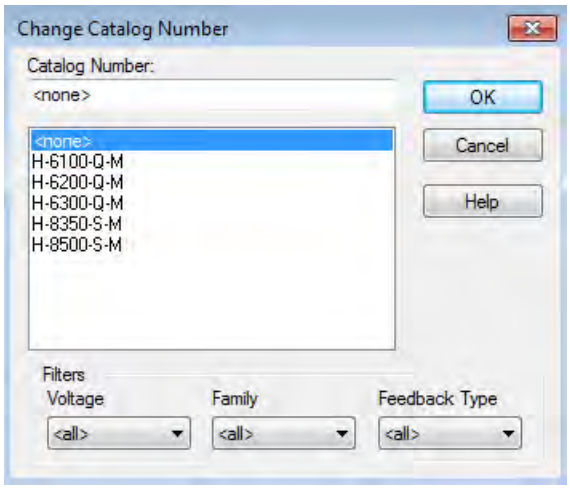


Item	Description
Amplifier Catalog Number	Select the catalog number of the amplifier to which this axis is connected.
Motor Catalog Number	Select the catalog number of the motor associated with this axis. When you change a Motor Catalog Number, the controller recalculates the values.

Change Catalog…	Lets you select a new motor catalog number. There are three fields that can be used to refine the selection process.

Drive/Motor Tab - (AXIS_SERVO_DRIVE) Recalculations

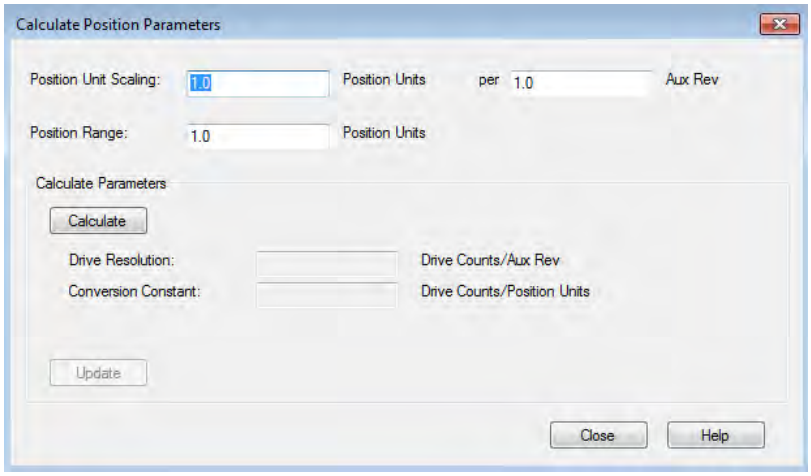
On this tab	**These attributes are recalculated**
Motor Feedback tab	Motor Feedback Type Motor Feedback Resolution
Gains tab	Position Proportional Gains Velocity Proportional Gains
Dynamics tab	Maximum Speed Maximum Acceleration Maximum Deceleration
Limits tab	Position Error Tolerance
Custom Stop Action Attributes dialog box	Stopping Torque
Custom Limit Attributes dialog box	Velocity Limit Bipolar Velocity Limit Positive Velocity Limit Negative Acceleration Limit Bipolar Acceleration Limit Positive Acceleration Limit Negative Torque Limit Bipolar Torque Limit Positive Torque Limit
Tune Bandwidth dialog box	Position Loop Bandwidth Velocity Loop Bandwidth
Drive/Motor Tab	
Loop Configuration	Select the configuration of the servo loop. • Motor Feedback Only – Displayed when Axis Configuration is Feedback only • Aux Feedback Only – Displayed when Axis Configuration is Feedback only • Position Servo • Aux Position Servo (not applicable to Ultra3000 drives) • Dual Position Servo • Dual Command Servo • Aux Dual Command Servo • Velocity Servo • Torque Servo • Dual Command/Feedback Servo



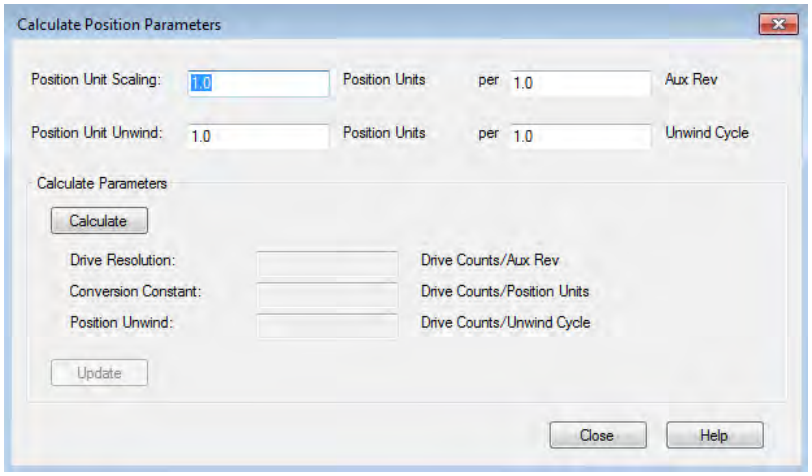
Item	Description	
Catalog Number	Lists the available catalog numbers based on the selection criteria from the fields in the **Filters** area.	
Filters	The following three optional fields let you refine your search of the Motor Database.	
	Voltage	Narrows the search to a voltage rating. The default is **all**.
	Family	Narrows the search to a family of motors. The default is **all**.
	Feedback Type	Narrows the search to a feedback type. The default is **all**.

When you click **Calculate** on the **Drive/Motor** tab for AXIS_SERVO_DRIVE, the **Calculate Position Parameters** dialog box opens.

When the **Conversion** tab has **Linear** as the value for **Position Mode**, clicking **Calculate** displays the following dialog box.



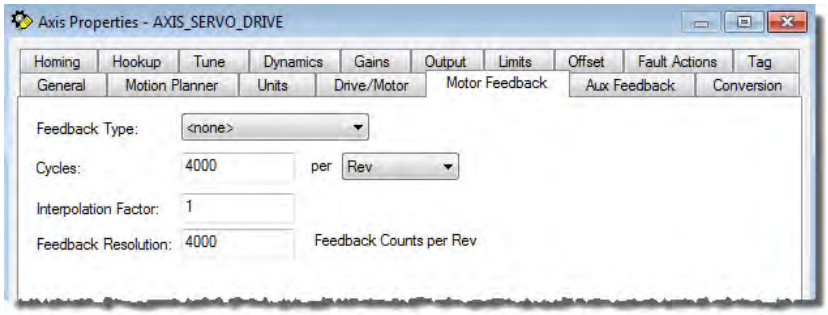
Position Unit Scaling	Position Unit Scaling defines the relationship between the Position Units defined on the **Units** tab and the units selected to measure position.
Per	The units used for Position Unit Scaling. The options are: Motor Inch, Motor Millimeter, or Motor Rev
Position Range	Maximum travel limit that your system can go.

Item	Description
Position Unit Unwind	For Rotary applications, the Position Unit Unwind field appears. Enter the value for the maximum number of unwinds in position units per unwind cycle.
Calculate Parameters	The Calculate Parameters shows the values that are to be calculated based upon the values entered for the **Position Unit Scaling** and **Position Range**.
Calculate	Opens a dialog box that lets you calculate the **Drive Resolution** and **Conversion Constant** based upon your input for **Position Unit Scaling** and **Position Range** for Linear Positioning mode. If you are in **Rotary Positioning** Mode, then it calculates the **Drive Resolution, Conversion Constant,** and **Position Unwind** based upon your inputs for **Position Unit Scaling** and **Position Unit Unwind**.
Drive Resolution	Recalculates the resolution based upon the new values entered on this dialog box.
Conversion Constant	Recalculates the **Conversion Constant** based upon the new values entered on this dialog box. When you edit the setting for the **Conversion Constant** or the **Drive Resolution**, and then click **OK** or **Apply**, you must choose whether to also recalculate the settings for these dependent attributes. The following attributes are recalculated. On the **Dynamics** tab: • Maximum Velocity • Maximum Acceleration • Maximum Deceleration On the **Limits** tab: • Position Error Tolerance On the **Custom Drive Scaling Attributes** dialog box: • Torque Data Scaling On the **Custom Limit Attributes** dialog box: • Velocity Limit Bipolar • Velocity Limit Positive • Velocity Limit Negative • Acceleration Limit Bipolar • Acceleration Limit Positive • Acceleration Limit Negative When the **Conversion** tab has **Rotary** as the value for **Position Mode**, clicking **Calculate** displays the following dialog box. 

<table>
<tr><th>Item</th><th>Description</th></tr>
<tr><td colspan="2">Drive/Motor Tab</td></tr>
<tr><td>Dive Resolution</td><td>Type the number of counts per motor revolution. This value applies to all position data. Valid values range from 1 to 2^32 - 1. One Least Significant Bit (LSB) for position data equals 360° / RotationalPositionResolution.
Note that the Drive Resolution is also referred to as Rotational Position Resolution.
When you save an edited Conversion Constant or a Drive Resolution value, a message box opens, asking you if you want the controller to automatically recalculate certain attribute settings. See Conversion Constant on page 199 for a list of these attributes.
Drive Resolution is especially helpful for fractional unwind applications or multi-turn applications requiring cyclic compensation. You can modify the Drive Resolution value so that dividing it by the Unwind Value yields a whole integer value. The higher the Drive Resolution setting, the finer the resolution.</td></tr>
<tr><td>Drive Counts per</td><td>Choose the units you want to use for this drive. Choose from Motor Inch, Motor Millimeter, and Motor Rev (default).</td></tr>
<tr><td>Drive Enable Input Checking</td><td>Check this box to enable Drive Enable Input Checking. When this box is checked, the drive regularly monitors the state of the Drive Enable Input. This dedicated input enables the drive's power structure and servo loop. If cleared, no such checking of the Drive Enable Input occurs.</td></tr>
<tr><td>Drive Enable Input Fault</td><td>Check this box to activate the Drive Enable Input Fault. When active, a fault detected on the external drive notifies the motion module via Drive Fault Input.</td></tr>
<tr><td>Real Time Axis Information Attribute 1/Attribute 2</td><td>Select up to two axis attributes whose statuses are transmitted – along with the actual position data – to the Logix processor. You can access the values of the selected attributes using a GSV command or from the axis tag itself. This data is transmitted at a rate equal to the servo status data update time.
If you issue a GSV command for servo status attribute, or use the value from the axis tag, without having selected this attribute via the Drive Info Select attribute, the attribute value is static and does not reflect the true value in the servo module.
If the AXIS_SERVO_DRIVE is associated with a Kinetix Enhanced Safe Torque-Off or Advanced Safety Drive, these two additional Real Time Axis attributes are available.
• Guard Status
• Guard Faults
If an AXIS_SERVO_DRIVE is associated with a Kinetix Advanced Safety Drive, we recommend that you configure the Guard Status attribute. Otherwise, you receive a warning when verifying your project.
<table>
<tr><th>If the AXIS_SERVO_DRIVE is associated with a Kinetix Advanced Safety Drive, and</th><th>Then</th></tr>
<tr><td>Attribute 1 or Attribute 2 is populated as Guard Status</td><td>No action is taken.</td></tr>
<tr><td>Attribute 1 or Attribute 2 is not populated as Guard Status</td><td>Attribute 2 is populated as Guard Status.</td></tr>
<tr><td>Attribute 2 is populated with an attribute other than Guard Status, and Attribute 1 is undefined</td><td>Attribute 1 is populated as Guard Status.</td></tr>
<tr><td>Attribute 1 and Attribute 2 are populated with an attribute other than Guard Status</td><td>Upon project verification, a warning is issued.</td></tr>
</table>
</td></tr>
</table>

Motor Feedback tab - AXIS_SERVO_DRIVE

Use the **Motor Feedback** tab to configure motor and auxiliary feedback device (if any) parameters, for an axis of the type AXIS_SERVO_DRIVE.

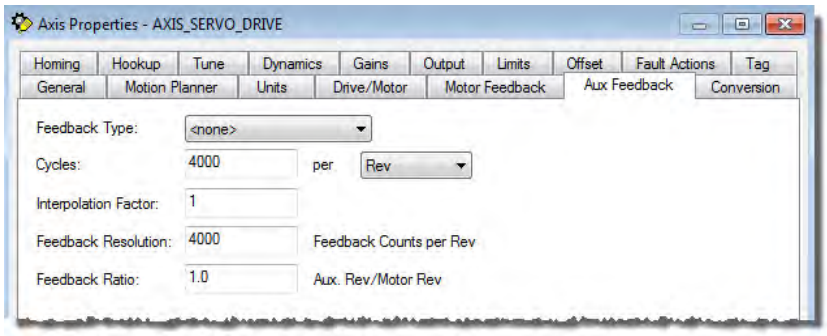


The **Axis Configuration** selection made on the **General** tab, and the **Loop Configuration** selection made on the **Drive** tab determine which sections of this dialog box – Motor and Auxiliary Feedback – are enabled.

Item	Description
Feedback Type	This field displays the type of feedback associated with the selected motor.
Cycles	The number of cycles of the associated feedback device. This helps the Drive Compute Conversion constant used to convert drive units to feedback counts. Depending on the feedback type you select, this value may be read-only or editable.
Per	The units used to measure the cycles.
Interpolation Factor	This field displays a fixed, read-only value for each feedback type. This value is used to compute the resolution of the feedback device.
Feedback Resolution	Provides the drive with the resolution of the associated feedback device in cycles.

Aux Feedback tab - AXIS_SERVO_DRIVE

The **Aux Feedback** tab is enabled only if on the **Driver** tab, the **Loop Configuration** field is set to **Aux Feedback Only**, **Aux Position Servo**, **Dual Position Servo**, **Dual Command Servo**, or **Aux Dual Command Servo**. Use this tab to configure motor and auxiliary feedback device parameters for an axis of the type AXIS_SERVO_DRIVE.

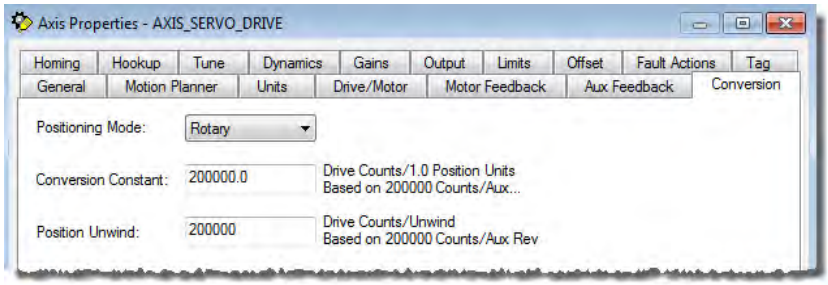


Item	Description
Feedback Type	For applications that use auxiliary feedback devices, select the type of auxiliary feedback device type. These are drive dependent.
Cycles	The number of cycles of the auxiliary feedback device. This helps the Drive Compute Conversion constant used to convert drive units to feedback counts. Depending on the feedback type selected, this value may be read-only or editable.
Per	The units used to measure the cycles.
Interpolation Factor	This field displays a fixed constant value for the selected feedback type. This value is used to compute the resolution of the feedback device.
Feedback Resolution	Provides the drive with the resolution of the associated feedback device in cycles.
Feedback Ratio	Represents the quantitative relationship between the auxiliary feedback device and the motor. Click the **Conversion** tab to access the **Axis Properties Conversion** dialog box.

Conversion tab

Use the **Conversion** tab to view and edit the **Positioning Mode**, **Conversion Constant**, and if configured as **Rotary**, the **Position Unwind** values for an axis, of the tag types AXIS_SERVO, AXIS_SERVO_DRIVE and AXIS_VIRTUAL.

The differences in the appearance of the **Conversion** tab for the AXIS_SERVO and AXIS_SERVO_DRIVE are the default values for **Conversion Constant** and **Position Unwind** and the labels for these values.

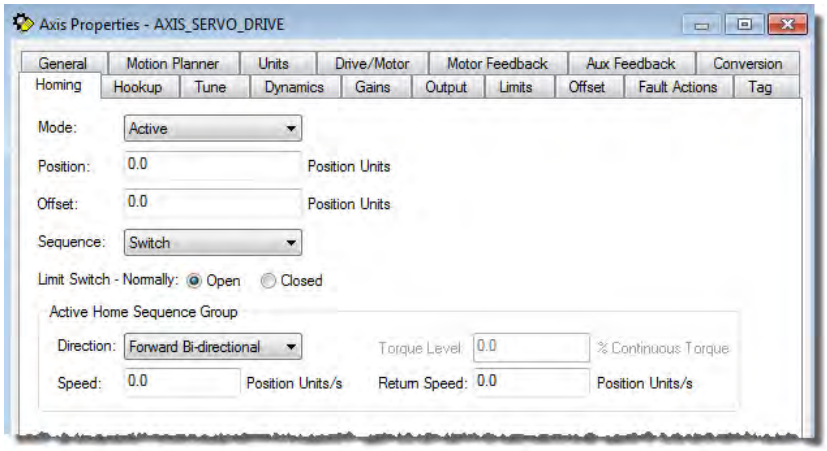


Item	Description	
Positioning Mode	This parameter is not editable for an axis of the data type AXIS_CONSUMED. Instead, this value is taken from a producing axis in a networked Logix processor. This value can be edited for AXIS_SERVO, AXIS_SERVO_DRIVE and AXIS_VIRTUAL.	
	Linear	Provides a maximum total linear travel of 2.14 (231) billion feedback counts. With this mode, the unwind feature is disabled and you can limit the linear travel distance of the axis by specifying the positive and negative travel limits for the axis.
	Rotary	Enables the rotary unwind capability of the axis. This feature provides infinite position range by unwinding the axis position whenever the axis moves through a complete unwind distance. The number of encoder counts per unwind of the axis is specified in the **Position Unwind** field.

Item	Description
Conversion Constant	Number of feedback counts per position unit. This conversion or 'K' constant lets the axis position display, and motion to be programmed, in the position units set in the **Units** tab. The conversion constant is used to convert axis position units into feedback counts and vice versa for the AXIS_SERVO type and for the AXIS_SERVO_DRIVE, the number of counts per motor revolution, as set on the **Drive** tab in the **Drive Resolution** field. When you edit the setting for the **Conversion Constant** or the **Drive Resolution** and click **OK** or **Apply**, you must choose whether to also recalculate the settings for these dependent attributes. The following attributes are recalculated. On the **Dynamics** tab: • Maximum Velocity • Maximum Acceleration • Maximum Deceleration On the **Limits** tab: • Position Error Tolerance On the **Custom Drive Scaling Attributes** dialog box: • Torque Data Scaling On the **Custom Limit Attributes** dialog box: • Velocity Limit Bipolar • Velocity Limit Positive • Velocity Limit Negative • Acceleration Limit Bipolar • Acceleration Limit Positive • Acceleration Limit Negative
Position Unwind	This parameter is not editable for an axis of the data type AXIS_CONSUMED. Instead, this value is taken from a producing axis in a networked Logix processor. For a Rotary axis (AXIS_SERVO): • This value represents the distance (in feedback counts) used to perform automatic electronic unwind. Electronic unwind allows infinite position range for rotary axes by subtracting the unwind distance from the actual and command position, every time the axis travels the unwind distance. For axes of the type AXIS_SERVO_DRIVE: • When you save an edited Conversion Constant or a Drive Resolution value, a message box opens, asking if you want the controller to automatically recalculate certain attribute settings. See Conversion Constant and Drive Resolution attributes. • The label indicates the number of counts per motor revolution, as set on the **Drive** tab in the **Drive Resolution** field.

Homing tab - AXIS_SERVO

Use the **Homing** tab to configure the attributes related to homing an axis of the type AXIS_SERVO.



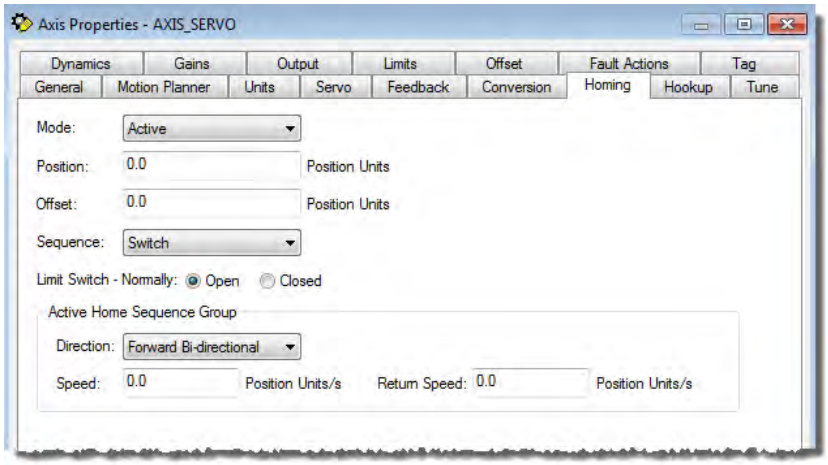
Item	Description	
Mode	Select the homing mode.	
	Active	• Active mode - enables the axis at the beginning of the home process. • Active homing sequences use the trapezoidal velocity profile. For LDT and SSI feedback selections, the only valid Home Sequences for Homing Mode are immediate or switch, as no physical marker exists for the LDT or SSI feedback devices. • For SSI, the selections for Home Sequence are based on if 'Enable Absolute Feedback' is checked.
	Passive	• The homing redefines the absolute position of the axis on the occurrence of a home switch or encoder marker event. • Passive homing is most commonly used to calibrate uncontrolled axes, although it can also be used with controlled axes to create a custom homing sequence. • Passive homing, for a given home sequence, works similar to the corresponding active homing sequence, except that no motion is commanded; the controller just waits for the switch and marker events to occur.
	Absolute	AXIS_SERVO_DRIVE, and AXIS_SERVO when associated with a 1756-HYD02 [LDT feedback] or 1756-M02AS [SSI feedback] module only. Absolute homing is only available on Axis_Servo_Drive if the position feedback devices support absolute homing. Absolute Homing is not available on a 1756-M02AS module if **Enable Absolute Feedback** is cleared. • The absolute homing process establishes the true absolute position of the axis by applying the configured Home Position to the reported position of the absolute feedback device. • The only valid Home Sequence for an absolute Homing Mode is immediate. In the LDT and SSI cases, the absolute homing process establishes the true absolute position of the axis by applying the configured Home Position less any enabled Absolute Feedback Offset to the reported position of the absolute feedback device. • Before execution of the absolute homing process using the MAH instruction, the axis must be in the Axis Ready state with the servo loop disabled.

<table>
<tr><th>Item</th><th colspan="2">Description</th></tr>
<tr><td></td><td colspan="2">No Physical Marker Pulse Exists
For the SSI feedback transducer, no physical marker pulse exists. However, a pseudo marker reference is established by the M02AS module firmware at the feedback device's roll over point. A single-turn Absolute SSI feedback device rolls over at its maximum 'turns count' = 1 rev. A multi-turn Absolute SSI feedback device (there are multiple revs or feedback-base unit-distances) rolls over at its maximum 'turns count', which is usually 1024 or 2048.
Home to Rollover is available in the Homing Properties if Enable Absolute Feedback is cleared.
If you must establish the rollover of the feedback device, a ladder rung using an SSV to set Home_Sequence equal 'Home to Rollover' is available in the Homing Properties if Enable Absolute Feedback is cleared.
The following parameters must be added to the application program.
• Class Name = Axis,
• Attribute_Name = Home_Sequence,
• and Value = 2 (to Marker)
These cannot be set in Axis Properties. They must be reset back to the initial values: 0 = Immediate or 1 = Switch, after establishing the rollover.
The Home Sequence = to Marker must be used to allow feedback to travel until the rollover (that is, pseudo marker) is found. This must be done without the motor attached to any axis as this could cause up to a Maximum number of turns before pseudo marker is found.</td></tr>
<tr><td>Position</td><td colspan="2">The desired absolute position, in position units, for the axis after the specified homing sequence has been completed. In most cases, this position is set to zero, although any value within the software travel limits can be used. After the homing sequence is complete, the axis is left in this position.
If the Positioning Mode (set in the Conversion tab) of the axis is Linear, then the home position should be within the travel limits, if enabled. If the Positioning Mode is Rotary, then the home position should be less than the unwind distance in position units.</td></tr>
<tr><td>Offset</td><td colspan="2">The desired offset (if any) in position units the axis is to move, upon completion of the homing sequence, to reach the home position. In most cases, this value is zero.</td></tr>
<tr><td>Sequence</td><td colspan="2">Select the event that causes the Home Position to be set. See the Homing Configurations section, for a detailed description of each combination of homing mode, sequence and direction.</td></tr>
<tr><td></td><td>Sequence Type</td><td>Description</td></tr>
<tr><td></td><td>Immediate</td><td>Sets the Actual Position to the Home Position.</td></tr>
<tr><td></td><td>Switch</td><td>Sets the Actual Position to the Home Position when axis motion encounters a home limit switch.</td></tr>
<tr><td></td><td>Marker</td><td>Sets the Actual Position to the Home Position when axis encounters an encoder marker.</td></tr>
<tr><td></td><td>Switch-Marker</td><td>Sets the Actual Position to the Home position when a marker is encountered after a home switch is encountered.</td></tr>
<tr><td>Limit Switch</td><td colspan="2">If a limit switch is used, indicates the normal state of that switch (that is, before being engaged by the axis during the homing sequence):
• Normally Open
• Normally Closed</td></tr>
<tr><td>Direction</td><td colspan="2">For active homing sequences, except for the Immediate Sequence type, select the desired homing direction.</td></tr>
<tr><td></td><td>Direction</td><td>Description</td></tr>
<tr><td></td><td>Forward Uni-directional</td><td>The axis jogs in the positive axial direction until a homing event (switch or marker) is encountered, then continues in the same direction until axis motion stops (after decelerating or moving the Offset distance).</td></tr>
<tr><td></td><td>Forward Bi-directional</td><td>The axis jogs in the positive axial direction until a homing event (switch or marker) is encountered, then reverses direction until motion stops (after decelerating or moving the Offset distance).</td></tr>
<tr><td></td><td>Reverse Uni-directional</td><td>The axis jogs in the negative axial direction until a homing event (switch or marker) is encountered, then continues in the same direction until axis motion stops (after decelerating or moving the Offset distance).</td></tr>
</table>

Item	Description	
	Reverse Bi-directional	The axis jogs in the negative axial direction until a homing event (switch or marker) is encountered, then reverses direction until motion stops (after decelerating or moving the Offset distance).
Speed	Type the speed of the jog profile used in the first leg of an active homing sequence. The homing speed specified should be less than the maximum speed and greater than zero.	
Return Speed	The speed of the jog profile used in the return leg(s) of an active homing sequence. The home return speed specified should be less than the maximum speed and greater than zero.	

Homing tab - AXIS_SERVO_DRIVE

Use the **Homing** tab to configure the attributes related to homing an axis of the type AXIS_SERVO_DRIVE.



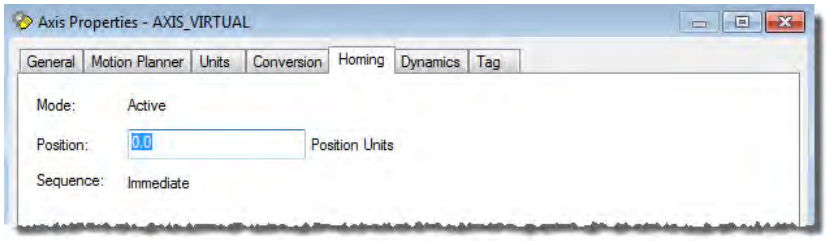
Item	Description	
Mode	Select the homing mode.	
	Active	• Active enables the axis at the beginning of the home process. • The desired homing sequence is selected by specifying whether a home limit switch and/or the encoder marker is used for this axis. • Active homing sequences use the trapezoidal velocity profile. For LDT and SSI feedback selections, the only valid Home Sequences for Homing Mode are immediate or switch, as no physical marker exists for the LDT or SSI feedback devices.

Item	Description	
	Passive	• In this mode, homing redefines the absolute position of the axis on the occurrence of a home switch or encoder marker event. • Passive homing is most commonly used to calibrate uncontrolled axes, although it can also be used with controlled axes to create a custom homing sequence. • Passive homing, for a given home sequence, works similar to the corresponding active homing sequence, except that no motion is commanded; the controller just waits for the switch and marker events to occur.
	Passive Absolute	AXIS_SERVO_DRIVE and AXIS_SERVO when associated with a 1756-HYD02 [LDT feedback] or 1756-M02AS [SSI feedback] module only. • In this mode, the absolute homing process establishes the true absolute position of the axis by applying the configured Home Position to the reported position of the absolute feedback device. • The only valid Home Sequence for an absolute Homing Mode is immediate. In the LDT and SSI cases, the absolute homing process establishes the true absolute position of the axis by applying the configured Home Position less any enabled Absolute Feedback Offset to the reported position of the absolute feedback device. • Before execution of the absolute homing process using the MAH instruction, the axis must be in the Axis Ready state with the servo loop disabled. **No Physical Marker Pulse Exists** For the SSI feedback transducer, no physical marker pulse exists. However, a pseudo marker reference is established by the M02AS module firmware at the feedback device's roll over point. A single-turn Absolute SSI feedback device rolls over at its maximum 'turns count' = 1 rev. A multi-turn Absolute SSI feedback device (there are multiple revs or feedback-base unit-distances) rolls over at its maximum 'turns count', which is usually 1024 or 2048. If you must establish the rollover of the feedback device, a ladder rung using an SSV to set Home_Sequence equal 'Home to marker' with the following parameters: Class Name = SSI_Axis, Attribute_Name = Home_Sequence, and Value = 2 (to Marker) must be added to the application program (cannot be set to Axis Properties and must be reset back to its initial value 0 = Immediate or 1 = Switch after establishing the rollover). The Home Sequence = to Marker must be used to allow feedback to travel until the rollover (that is, pseudo marker) is found. This must be done without the motor attached to any axis as this could cause up to the Maximum number of turns before pseudo marker is found.
	Position	Type the desired absolute position, in position units, for the axis after the specified homing sequence has been completed. In most cases, this position is set to zero, although any value within the software travel limits can be used. After the homing sequence is complete, the axis is left in this position. If the **Positioning mode** (set in the **Conversion tab**) of the axis is **Linear**, then the home position should be within the travel limits, if enabled. If the **Positioning mode** is **Rotary**, then the home position should be less than the unwind distance in position units.
	Offset	Type the desired offset (if any) in position units the axis is to move, upon completion of the homing sequence, to reach the home position. In most cases, this value is zero.
Sequence	Select the event that causes the Home Position to be set.	
	Immediate	Sets the Actual Position to the Home Position.
	Switch	Sets the Actual Position to the Home Position when axis motion encounters a home limit switch.
	Marker	Sets the Actual Position to the Home Position when axis encounters an encoder marker.
	Switch-Marker	Sets the Actual Position to the Home position when a marker is encountered after a home switch is encountered.

Item	Description	
	Torque Level	Sets the Home Position when the specified Homing Torque level is achieved on the assigned axis. **Important:** For more information on the Home to Torque-level sequence, see Home to Torque-level Example Application Note, publication MOTION-AT001.
	Torque Level-marker	Sets the Home Position when the specified Homing Torque level is achieved on the assigned axis, only after the axis encounters an encoder marker. **Important:** For more information on the Home to Torque-level sequence, see Home to Torque-level Example Application Note, publication MOTION-AT001. See the section Homing Configurations, for a detailed description of each combination of homing mode, sequence and direction.
Limit Switch	If a limit switch is used, indicate the normal state of that switch (that is, before being engaged by the axis during the homing sequence). • Normally Open • Normally Closed	
Direction	For active homing sequences, except for the Immediate Sequence type, select the desired homing direction	
	Direction	**Description**
	Forward Uni-directional	The axis jogs in the positive axial direction until a homing event (switch or marker) is encountered, then continues in the same direction until axis motion stops (after decelerating or moving the Offset distance).
	Forward Bi-directional	The axis jogs in the positive axial direction until a homing event (switch or marker) is encountered, then reverses direction until motion stops (after decelerating or moving the Offset distance).
	Reverse Uni-directional	The axis jogs in the negative axial direction until a homing event (switch or marker) is encountered, then continues in the same direction until axis motion stops (after decelerating or moving the Offset distance).
	Reverse Bi-directional	The axis jogs in the negative axial direction until a homing event (switch or marker) is encountered, then reverses direction until motion stops (after decelerating or moving the Offset distance).
Speed	Type the speed of the jog profile used in the first leg of an active homing sequence. The homing speed specified should be less than the maximum speed and greater than zero.	
Torque Level	The torque level, with units % continuous torque, that the axis motor must reach to complete the Home-to-Torque sequence. This feature is only available on the Kinetix family of drives.	
Return Speed	The speed of the jog profile used in the return leg(s) of an active homing sequence. The home return speed specified should be less than the maximum speed and greater than zero.	

Homing tab - AXIS_VIRTUAL

Use the **Homing** tab to configure the attributes related to homing an axis of the type AXIS_VIRTUAL.



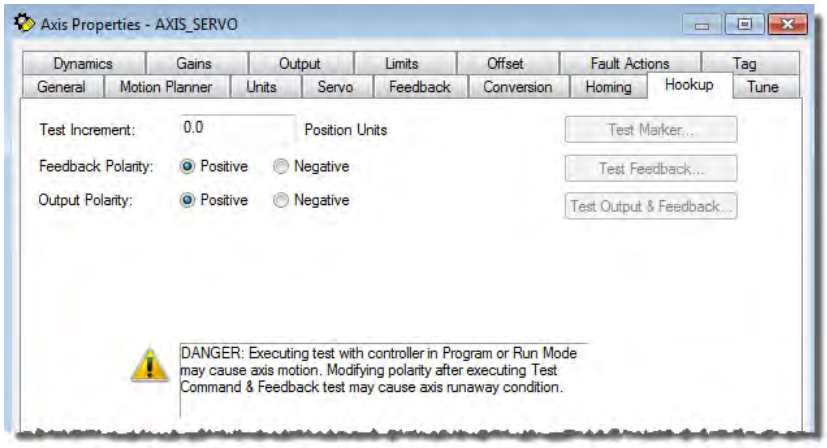
Only an Active Immediate Homing sequence can be performed for an axis of the type AXIS_VIRTUAL. A virtual axis is always enabled. The controller assigns the

Home Position to the current axis actual position and command position. This homing sequence produces no axis motion.

Item	Description
Mode	This read-only parameter is set to Active.
Position	Desired absolute position, in position units, for the axis after the specified homing sequence has been completed. In most cases, this position is set to zero, although any value within the software travel limits can be used. After the homing sequence is complete, the axis is left at this position. If the **Positioning Mode** (set in the **Conversion** tab) of the axis is **Linear**, then the home position should be within the travel limits, if enabled. If the **Positioning Mode** is **Rotary**, then the home position should be less than the unwind distance in position units.
Sequence	This read-only parameter is set to Immediate.

Hookup tab - AXIS_SERVO

Use the **Hookup** tab to configure and initiate axis hookup and marker test sequences for an axis of the type AXIS_SERVO.



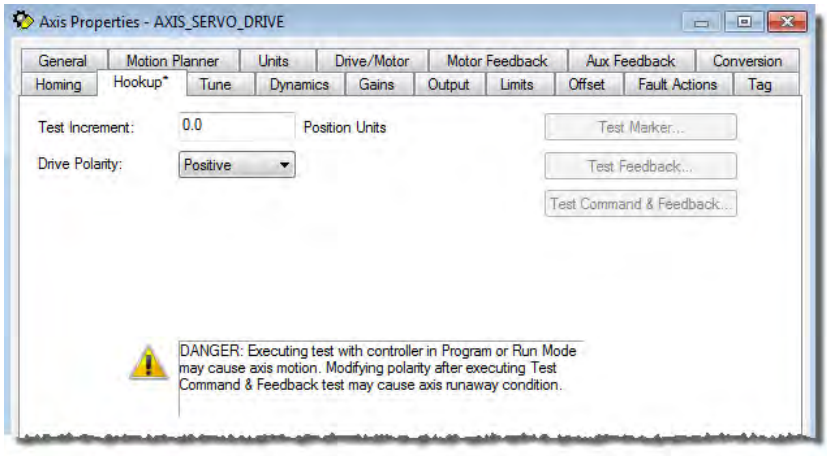
When a parameter transitions to a read-only state, any pending changes to parameter values are lost, and the parameter reverts to the most recently saved parameter value.

Item	Description
Test Increment	Specifies the amount of distance traversed by the axis when executing the Output & Feedback test. Test Increment is also used for the Marker and Feedback test. The test is complete when the distance is traveled. For example, if the distance is set to 1/4 of the revolution, then the marker test will fail 75% of the time because the marker will never be seen. For the Marker test, the test increment has to be a distance large enough to ensure that a marker is passed. The default value is set to approximately a quarter of a revolution of the motor in position units.

Item	Description
Feedback Polarity	The polarity of the encoder feedback, this field is automatically set by executing the Feedback Test or the Output & Feedback Test. This field is set only after the test is executed and the user accepts the results. • Positive • Negative When properly configured, this setting ensures that axis Actual Position value increases when the axis is moved in the user defined positive direction. This bit can be configured automatically using the MRHD and MAHD motion instructions.
	WARNING: Modifying input polarity values by running the Feedback or Output & Feedback Tests can cause an unexpected motion resulting in damage to the equipment, and physical injury or death.
Output Polarity	The polarity of the servo output to the drive, this field is automatically set by executing and accepting the results of the Output & Feedback Test. • Positive • Negative When properly configured, this setting and the **Feedback Polarity** setting ensure that, when the axis servo loop is closed, it is closed as a negative feedback system and not an unstable positive feedback system. This bit can be configured automatically using the MRHD and MAHD motion instructions.
Test Marker	Runs the Marker test, which ensures that the encoder A, B, and Z channels are connected correctly and phased properly for marker detection. When the test is initiated, you must manually move the axis the distance specified by the Travel Limit for the system to detect the marker. If the marker is not detected, check the encoder wiring and try again.
Test Feedback	Runs the Feedback Test, which checks and, if necessary, reconfigures the **Feedback Polarity** setting. When the test is initiated, you must manually move the axis one revolution for the system to detect the marker. If the marker is not detected, check the encoder wiring and try again.
Test Output & Feedback	Runs the Output & Feedback Test, which checks and, if necessary, reconfigures the polarity of encoder feedback (the **Feedback Polarity** setting) and the polarity of the servo output to the drive (the **Output Polarity** setting), for an axis configured for Servo operation in the **General** tab. Executing and accepting the values automatically saves all changes to Axis Properties. It is possible to execute the test, but not accept (or apply) the values.

Hookup tab - AXIS_SERVO_DRIVE

Use this tab to configure and initiate axis hookup and marker test sequences for an axis of the type AXIS_SERVO_DRIVE.



When a parameter transitions to a read-only state, any pending changes to parameter values are lost, and the parameter reverts to the most recently saved parameter value.

Test Increment

Specifies the amount of distance traversed by the axis when executing the Command & Feedback test. The default value is set to approximately a quarter of a revolution of the motor in position units.

Drive Polarity

The polarity of the servo loop of the drive, set by executing the Command & Feedback Test.

- Positive
- Negative

Proper wiring guarantees that the servo loop is closed with negative feedback. However, there is no guarantee that the drive has the same sense of forward direction as the user for a given application. Negative Polarity inverts the polarity of the command position and actual position data of the drive. Thus, selecting Positive or Negative Drive Polarity makes it possible to configure the positive direction sense of the drive to agree with that of the user. This attribute can be configured automatically using the MRHD and MAHD motion instructions.

ATTENTION: Modifying polarity values, automatically input by running the Command & Feedback Test, can cause an unexpected motion.

Test Marker

Runs the Marker test, which ensures that the encoder A, B, and Z channels are connected correctly and phased properly for marker detection. When the test is initiated, you must manually move the axis one revolution for the system to detect the marker. If the marker is not detected, check the encoder wiring and try again.

Test Feedback

Runs the Feedback Test, which checks and, if necessary, reconfigures the Feedback Polarity setting. When the test is initiated, you must manually move the axis one revolution for the system to detect the marker. If the marker is not detected, check the encoder wiring and try again.

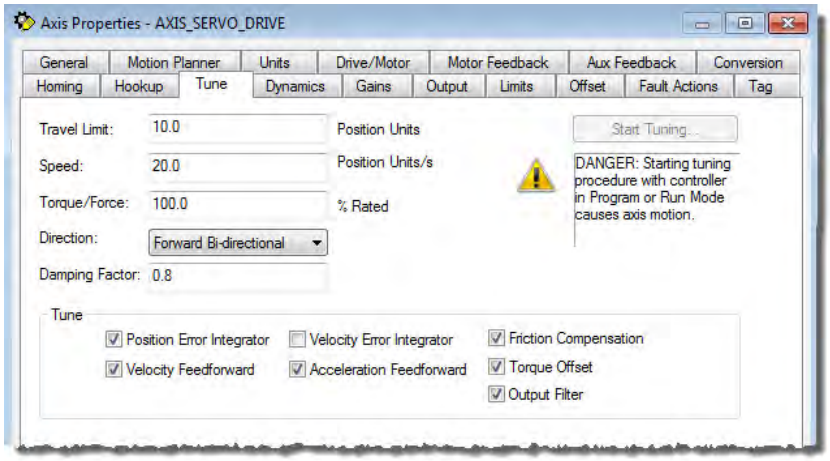
Test Command & Feedback

Runs the Command & Feedback Test, which checks and, if necessary, reconfigures the polarity of encoder feedback (the **Feedback Polarity** setting) and the polarity of the servo output to the drive (the **Output Polarity** setting), for an axis configured for Servo operation in the **General** tab.

Executing any test operation automatically saves all changes to axis properties.

Tune tab - AXIS_SERVO, AXIS_SERVO_DRIVE

Use this tab to configure and initiate the axis tuning sequence for an axis of the types AXIS_SERVO or AXIS_SERVO_DRIVE.



Travel Limit

Specifies a limit to the excursion of the axis during the tune test. If the servo module determines that the axis is not able to complete the tuning process before exceeding the tuning travel limit, it terminates the tuning profile and report that this limit was exceeded.

Speed

Determines the maximum speed for the tune process. This value should be set to the desired maximum operating speed of the motor (in engineering units) before running the tune test.

Torque/Force (AXIS_SERVO_DRIVE)

The maximum torque of a Rotary motor, or Force, for a linear motor. Force is used only when a linear motor is connected to the application. This attribute should be set to the desired maximum safe torque level before running the tune test. The default value is 100%, which yields the most accurate measure of the acceleration and deceleration capabilities of the system.

In some cases, a lower tuning torque limit value may be desirable to limit the stress on the mechanics during the tuning procedure. In this case, the acceleration and deceleration capabilities of the system are extrapolated based on the ratio of the tuning torque to the maximum torque output of the system. Extrapolation error increases as the Tuning Torque value decreases.

Torque (AXIS_SERVO)

The maximum torque of the tune test. This attribute should be set to the desired maximum safe torque level before running the tune test. The default value is 100%, which yields the most accurate measure of the acceleration and deceleration capabilities of the system.

In some cases, a lower tuning torque limit value may be desirable to limit the stress on the mechanics during the tuning procedure. In this case, the acceleration and deceleration capabilities of the system are extrapolated based on the ratio of the tuning torque to the maximum torque output of the system. Extrapolation error increases as the Tuning Torque value decreases.

Direction

The direction of the tuning motion profile.

- **Forward Uni-directional** – Initiated in the forward tuning direction only.
- **Forward Bi-directional** – First initiated in the forward tuning direction and then, if successful, is repeated in the reverse direction. Information returned by the Bi-directional Tuning profile can be used to tune Backlash Compensation and Torque Offset.

- **Reverse Uni-directional** – Initiated in the reverse tuning direction only.
- **Reverse Bi-directional** – First initiated in the reverse tuning direction and then, if successful, is repeated in the forward direction. Information returned by the Bi-directional Tuning profile can be used to tune Backlash Compensation and Torque Offset.

Damping Factor

Specifies the dynamic response of the servo axis. The default is set to 0.8. When gains are tuned using a small damping factor, a step response test performed on the axis may generate uncontrolled oscillation. The gains generated using a larger damping factor would produce a system step response that has no overshoot and is stable, but may be sluggish in response to changes.

The tuning procedure uses the Damping Factor that is set in this field. However, when the controller recalculates certain attributes in response to a Motor Catalog Number change (on the **Motor/Feedback** tab), the controller uses the default Damping Factor value of 0.8, and not another value set in this field.

Tune

Select the gains to be determined by the tuning test.

- **Position Error Integrator** – Determines whether to calculate a value for the Position Integral Gain.
- **Velocity Feedforward** – Determines whether to calculate a value for the Velocity Feedforward Gain.
- **Velocity Error Integrator** – Determines whether to calculate a value for the Velocity Integral Gain.
- **Acceleration Feedforward** – Determines whether to calculate a value for the Acceleration Feedforward Gain.
- **Backlash Compensation** – Determines whether to calculate a value for the Backlash Compensation Gain.
- **Torque Offset** – Determines whether to calculate a value for the Torque Offset. This tuning configuration is only valid if configured for bidirectional tuning.
- **Output Filter** – Determines whether to calculate a value for the Output Filter Bandwidth.

Start Tuning

Once the tune process completes successfully, you are prompted to accept the values. If the tuning process completes successfully, the following attributes are set. It is possible to complete the tuning process successfully and not accept (apply) the value (changes); therefore, the attributes are not set.

On this tab	These attributes are set
Gains tab	Velocity Feedforward Gain, if selected. Acceleration Feedforward Gain, if selected. Position Proportional Gain Position Integral Gain, if selected. Velocity Proportional Gain/Velocity Integral Gain, if selected. If cleared, the values are set to zero.
Dynamics tab	Maximum Speed Maximum Acceleration Maximum Deceleration Maximum Acceleration Jerk Maximum Deceleration Jerk
Output tab	Torque Scaling Velocity Scaling (AXIS_SERVO only) Low Pass Output Filter
Limits	Position Error Tolerance

The **Tune Bandwidth** dialog box opens for drives in which you can tweak bandwidth values.

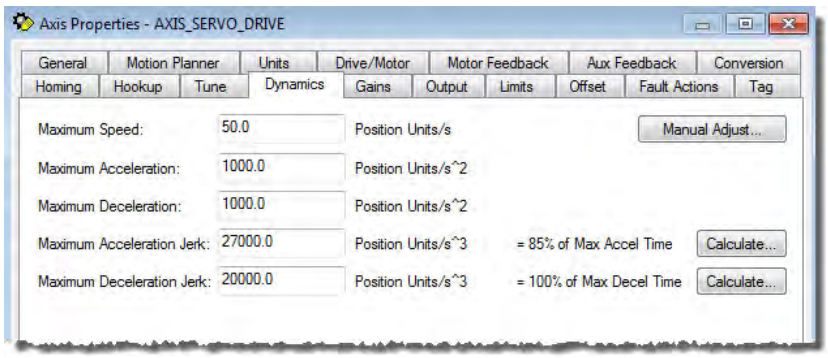
During tuning, if the controller detects a high degree of tuning inertia, it enables the Low Pass Output Filter and calculates and sets a value for Low Pass Output Filter Bandwidth.

Executing a Tune operation automatically saves all changes, only if the tune values are applied, to axis properties.

ATTENTION: This tuning procedure may cause axis motion with the controller in program mode. Unexpected motion may cause damage to the equipment, personal injury, or death.

Dynamics tab - AXIS_SERVO, AXIS_SERVO _DRIVE, AXIS_VIRTUAL

Use the **Dynamics** tab to view or edit the dynamics related parameters for an axis of the type AXIS_SERVO or AXIS_SERVO_DRIVE that is configured on the **General** tab for Servo operations, or for the type AXIS_VIRTUAL.



Important: The parameters on this tab can be edited in the following ways.

- Use this tab to edit parameter changes and then click **OK** to save your edits.
- Click **Manual Adjust**. Many attributes cannot be changed when online and/or the axis is enabled. Use Manual Adjust to make modifications to these attributes when online and the axis is enabled. Your changes are saved the moment a spin control changes any parameter value.

The parameters on this tab become read-only and cannot be edited when the controller is online if the controller is set to Hard Run mode, or if a Feedback On condition exists.

When the Logix Designer application is offline, the following parameters can be edited and the program saved to disk using the **Save** command or when you click **Apply**. You must re-download the edited program to the controller before it can be run.

Maximum Speed

The steady-state speed of the axis, it is initially set to Tuning Speed by the tuning process. This value is typically set to about 90% of the maximum speed rating of the motor. This provides sufficient 'head-room' for the axis to operate at all times within the speed limitations of the motor. The Maximum Speed value entered is used when the motion instruction is set with Speed Units = % of Maximum. If a motion instruction has a Speed Units = units per sec value entered, then the speed is taken from the motion instruction faceplate.

Maximum Acceleration

The maximum acceleration rate of the axis, in Position Units/second, it is initially set to about 85% of the measured tuning acceleration rate by the tuning process. If set manually, this value should typically be set to about 85% of the maximum acceleration rate of the axis. This provides sufficient 'head-room' for the axis to operate at all times within the acceleration limits of the drive and motor. The Maximum Acceleration value entered is used when the motion instruction is set with Accel Units = % of Maximum. When a motion instruction is configured with Accel Units = units per sec2 field, then the Maximum Acceleration is taken from the motion instruction faceplate.

Maximum Deceleration

The maximum deceleration rate of the axis, in Position Units/second, it is initially set to approximately 85% of the measured tuning deceleration rate by the tuning process. If set manually, this value should typically be set to about 85% of the maximum deceleration rate of the axis. This provides sufficient 'head-room' for the axis to operate at all times within the deceleration limits of the drive and motor. The Maximum Deceleration value entered is used when the motion instruction is set with decel Units=% of Maximum. When a motion instruction is configured with Decel Units=units per sec2 field, then the Maximum Deceleration is taken from the motion instruction faceplate.

Maximum Acceleration Jerk

The jerk parameters only apply to S-curve profile moves using the following instructions.

- MAJ
- MAM
- MAS
- MCD

The Maximum Acceleration Jerk rate of the axis, in Position Units/second3, defaults to 100% of the maximum acceleration time after tuning. The speed and acceleration rate for this calculation are determined during S-curve the tuning process.

$$\frac{\text{MaxAccel}^2}{\text{Speed}} = \text{Maximum Acceleration Jerk}$$

The Maximum Accel Jerk value entered is used when the motion instruction is set with Jerk Units=% of Maximum. When a Single-axis Motion Instruction has Jerk Units=units per sec3, then the maximum acceleration jerk value is derived from the motion instruction faceplate. The jerk units for the motion instruction also allow for Jerk Units=% of Time, with 100% of Time. This means that the entire S-curve move will have Jerk limiting. This is the default mode. An S-curve move with 0% of Time will result in a trapezoidal profile, and have 0% Jerk limiting. If set manually, enter the value in units=Position Units/second3 units. You can also use Calculate to view this value in terms of units=% of Time.

Maximum Deceleration Jerk

The jerk parameters only apply to S-curve profile moves using the following instructions.

- MAJ
- MAM
- MAS
- MCD

The Maximum Deceleration Jerk rate of the axis, in Position Units/second3, defaults to 100% of the maximum deceleration time after tuning. The speed and deceleration rate for the calculation are determined during the tuning process.

$$\frac{MaxDecel^2}{Speed} = \text{Maximum Deceleration Jerk}$$

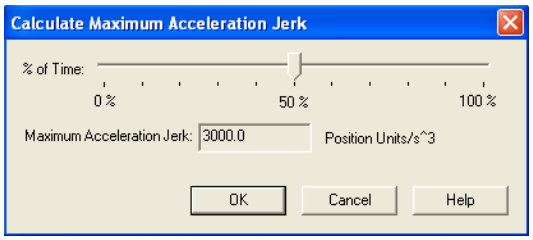
The Maximum Decel Jerk value entered is used when the motion instruction is set with Jerk Units=% of Maximum. When a Single-axis motion instruction has Jerk Units=units per sec3, then the Max Deceleration Jerk value is derived from the Motion Instruction faceplate. The jerk units for the motion instruction also allow for Jerk Units=% of Time, with 100% of Time meaning the entire S-curve move will have Jerk limiting, which is the default mode. An S-curve move with 0% of Time will result in a trapezoidal profile, and have 0% Jerk limiting. If set manually, enter the value in units=Position Units/second3 units. You can also use the optional Calculate to view the value in terms of units=% of Time.

Calculate button

This dialog box lets you set and view the Maximum Acceleration or Deceleration Jerk in Jerk Units=% of Time. Use the slider to select the value unit=% of Time. The numeric value in the **Maximum Accel\Decel Jerk** status box updates as the

slider is moved. Click **OK** to accept the new value, or click **Cancel** to exit without changing the value.

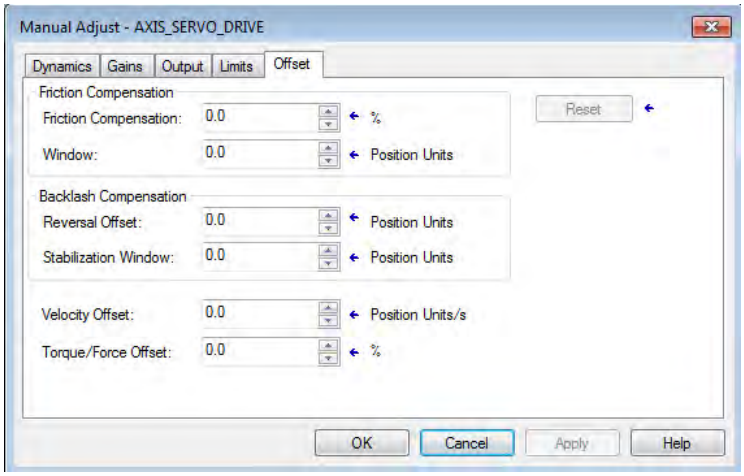
The Unit=% of Time is allowed for Jerk limiting only via the Instruction Faceplate. Only the Profile=S-curve allows Jerk control (Programmable S-curve). The units for programming Jerk limiting are more easily expressed in terms of % of Time rather than Position Units/s3.



Manual Adjust

Use **Dynamics** tab for online editing of the **Maximum Speed**, **Maximum Acceleration**, **Maximum Deceleration**, **Maximum Acceleration Jerk**, and **Maximum Deceleration Jerk**.

When values are changed on this dialog box, manually using the spin control or entering numeric values, a blue arrow appears. This means that the values have been instantaneously sent to the controller.

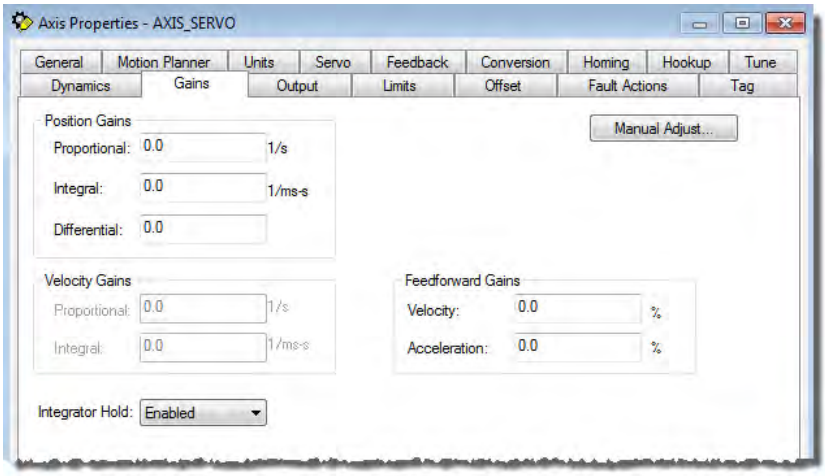


Manual Adjust is unavailable when the Logix Designer application is in Wizard mode, and when offline edits to the parameters have not yet been saved or applied.

Gains tab - AXIS_SERVO

Use the **Gains** tab to perform these offline functions for an axis of the type AXIS_SERVO, which is configured for Servo operations (set on the **General** tab of this dialog box), with Position Loop Configuration.

- Adjust gain values that are automatically set by the tuning process.
- Manually configure gains for the velocity and position loops.



The drive module uses a nested digital servo control loop consisting of a position loop with proportional, integral, and feed-forward gains around an optional digitally synthesized inner velocity loop.

Click **Manual Adjust** to edit parameter settings. Values with a blue arrow are sent to the controller. Click **Manual Adjust** to modify values when online and the axis is enabled. When online and the axis is enabled, the gain boxes on this dialog box are dimmed. The parameters on this tab become read-only and cannot be edited when the controller is online if the controller is set to Run mode, or if a Feedback On condition exists.

When Logix Designer application is offline, the following parameters can be edited and the program saved. You must download the edited program to the controller before it can be run.

Proportional (Position) Gain

Position Error is multiplied by the Position Loop Proportional Gain, or Pos P Gain, to provide a component to the Velocity Command that ultimately attempts to correct for the position error. Too little Pos P Gain results in excessively compliant, or mushy, axis behavior. Too large a Pos P Gain, on the other hand, can result in axis oscillation due to classical servo instability.

To set the gain manually, you must first set the output scaling factor (the **Velocity Scaling** factor or **Torque Scaling** factor) in the **Output** tab of this dialog box. Your selection of **External Drive Configuration** type,Torque or Velocity, in the **Servo** tab of this dialog box determines which scaling factor you must configure before manually setting gains.

If you know the desired loop gain in inches per minute per mil or millimeters per minute per mil, use the following formula to calculate the corresponding P gain:

Pos P Gain = 16.667 * Desired Loop Gain (IPM/mil)

If you know the desired unity gain bandwidth of the position servo in Hertz, use the following formula to calculate the corresponding P gain:

Pos P Gain = Bandwidth (Hertz) * 6.28

The typical value for the Position Proportional Gain is ~100 Sec-1.

Integral (Position) Gain

The Integral (that is, summation) of Position Error is multiplied by the Position Loop Integral Gain, or Pos I Gain, to produce a component to the Velocity Command that ultimately attempts to correct for the position error. Pos I Gain improves the steady-state positioning performance of the system. Increasing the integral gain generally increases the ultimate positioning accuracy of the system. Excessive integral gain, however, results in system instability.

In certain cases, Pos I Gain control is disabled. One such case is when the servo output to the axis drive is saturated. Continuing integral control behavior in this case would only exacerbate the situation. When the **Integrator Hold** parameter is set to **Enabled**, the servo loop automatically disables the integrator during commanded motion.

While the Pos I Gain, if employed, is typically established by the automatic servo tuning procedure (in the **Tuning** tab of this dialog box), the Pos I Gain value may also be set manually. However, it must be stressed that the Output Scaling factor for the axis must be established for the drive system. Once this is done, the Pos I Gain can be computed based on the current or computed value for the Pos P Gain using the following formula:

Pos I Gain = .025 * 0.001 Sec/mSec * (Pos P Gain)2

Assuming a Pos P Gain value of 100 Sec-1, this results in a Pos I Gain value of 2.5 ~0.1 mSec-1 - Sec-1.

Differential

Position Differential Gain helps predict a large overshoot before it happens and attempts to correct it before the overshoot actually occurs.

Proportional (Velocity) Gain

Velocity Error is multiplied by the Velocity Proportional Gain to provide a component to the Servo Output or Torque Command that ultimately attempts to correct for the velocity error, creating a damping effect. Thus, increasing the Velocity Proportional Gain results in smoother motion, enhanced acceleration, reduced overshoot, and greater system stability. However, too much Velocity Proportional Gain leads to high frequency instability and resonance effects.

If you know the desired unity gain bandwidth of the velocity servo in Hertz, you can use the following formula to calculate the corresponding P gain.

Velocity P Gain = Bandwidth (Hertz) / 6.28

The typical value for the Velocity Proportional Gain is 250.

Integral (Velocity) Gain

This parameter is enabled for all loop types except Torque loop.

At every servo update, the current Velocity Error is accumulated in a variable called the Velocity Integral Error. This value is multiplied by the Velocity Integral Gain to produce a component to the Servo Output or Torque Command that attempts to correct for the velocity error. The higher the Vel I Gain value, the faster the axis is driven to the zero Velocity Error condition. Unfortunately, I Gain control is intrinsically unstable. Too much I Gain results in axis oscillation and servo instability.

In certain cases, Vel I Gain control is disabled. One such case is when the servo output to the axis' drive is saturated. Continuing integral control behavior in this case would only exacerbate the situation. When the **Integrator Hold** parameter is set to **Enabled**, the servo loop automatically disables the integrator during commanded motion.

Due to the destabilizing nature of Integral Gain, it is recommended that Position Integral Gain and Velocity Integral Gain be considered mutually exclusive. If Integral Gain is needed for the application, use one or the other. In general, where static positioning accuracy is required, Position Integral Gain is the better choice.

The typical value for the Velocity Proportional Gain is ~15 mSec-2.

Velocity Feedforward

Velocity Feedforward Gain scales the current Command Velocity by the Velocity Feedforward Gain and adds it as an offset to the Velocity Command. Hence, the Velocity Feedforward Gain allows the following error of the servo system to be reduced to nearly zero when running at a constant speed. This is important in applications such as electronic gearing, position camming, and synchronization applications, where it is necessary that the actual axis position not significantly lag behind the commanded position at any time. The optimal value for Velocity Feedforward Gain is 100%, theoretically. In reality, however, you may need to update the value to accommodate velocity loops with non-infinite loop gain and other application considerations.

Acceleration Feedforward

Acceleration Feedforward Gain scales the current Command Acceleration by the Acceleration Feedforward Gain and adds it as an offset to the Servo Output generated by the servo loop. With this done, the servo loops do not contribute much to the Servo Output and the Position and/or Velocity Error values are significantly reduced. Therefore, when used in conjunction with the Velocity Feedforward Gain, the Acceleration Feedforward Gain allows the following error of the servo system during the acceleration and deceleration phases of motion to be reduced to nearly zero. This is important in applications such as electronic gearing, position camming, and synchronization applications, where it is necessary that the actual axis position not significantly lag behind the commanded position at any time. The optimal value for Acceleration Feedforward is 100%, theoretically. In reality, however, you may need to update the value to accommodate velocity loops with non-infinite loop gain and other application considerations.

Integrator Hold

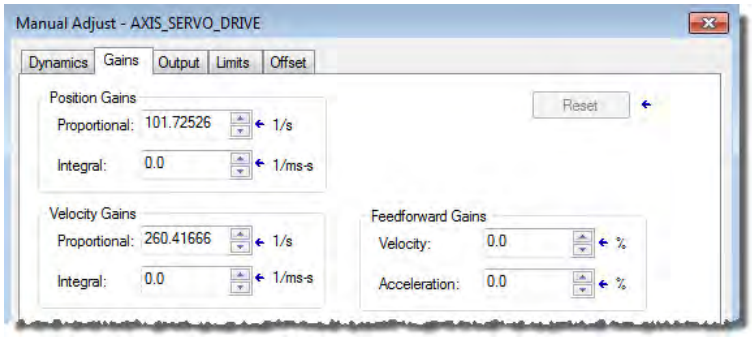
If the Integrator Hold parameter is:

- Checked, the servo loop temporarily stops any enabled position or velocity integrators while the command position is changing. This feature is used by point-to-point moves to minimize the integrator wind-up during motion.
- Unchecked, all active position or velocity integrators enabled.
- Enabled, the servo loop temporarily disables any enabled position or velocity integrators while the command position is changing. This feature is used by point-to-point moves to minimize the integrator wind-up during motion.

- Disabled, all active position or velocity integrators are enabled.

Manual Adjust

Opens the **Gains** tab of the **Manual Adjust** dialog box for online editing.

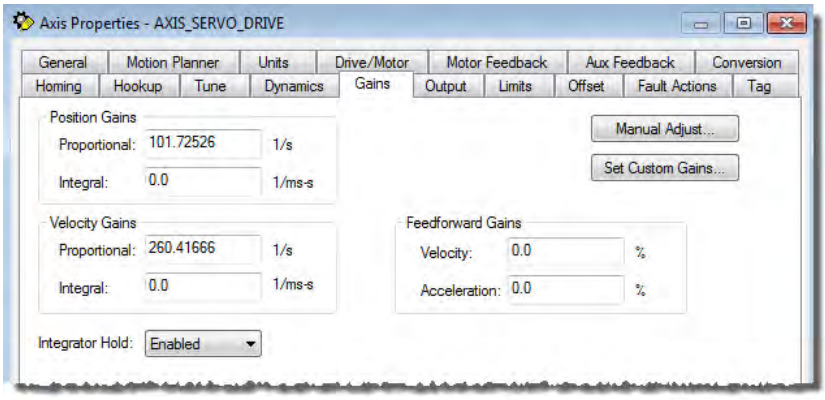


Manual Adjust is unavailable when Logix Designer application is in Wizard mode, and when you have not yet saved or applied your offline edits to the parameters.

Gains Tab - AXIS_SERVO_DRIVE

Use this tab to perform the following offline functions for an axis of the type AXIS_SERVO_DRIVE.

- Adjust, or tweak, gain values that are automatically set by the tuning process (in the **Tune** tab of this dialog box)
- Manually configure gains for the velocity and position loops



The drive module uses a nested digital servo control loop consisting of a position loop with proportional, integral, and feed-forward gains around an optional digitally synthesized inner velocity loop. The specific design of this nested loop depends upon the **Loop Configuration** selected in the **Drive** tab. For a discussion, including a diagram, of a loop configuration, click the following loop configuration types:

- Motor Position Servo Loop
- Auxiliary Position Servo Loop
- Dual Position Servo Loop
- Motor Dual Command Servo Loop
- Auxiliary Dual Command Servo Loop
- Velocity Servo Loop
- Torque Servo Loop

The parameters on this tab can be edited in the following ways.

- Edit on this tab by typing your parameter changes and then clicking **OK** to save your edits.
- Edit in the **Manual Adjust** dialog box: click **Manual Adjust** to open the **Manual Adjust** dialog box to this tab and use the spin controls to edit parameter settings. Your changes are saved the moment a spin control changes any parameter value.

The parameters on this tab become read-only and cannot be edited when the controller is online if the controller is set to Hard Run mode, or if a Feedback On condition exists.

When Logix Designer application is offline, the following parameters can be edited and the program saved to disk using the Save command or by clicking **Apply**. You must re-download the edited program to the controller before it can be run.

Velocity Feedforward

Velocity Feedforward Gain scales the current command velocity (derivative of command position) by the Velocity Feedforward Gain and adds it as an offset to the Velocity Command. Hence, the Velocity Feedforward Gain allows the following error of the servo system to be reduced to nearly zero when running at a constant speed. This is important in applications such as electronic gearing and synchronization applications, where it is necessary that the actual axis position not significantly lag behind the commanded position at any time. The optimal value for Velocity Feedforward Gain is 100%, theoretically. In reality, however, you may need to update the value to accommodate velocity loops with non-infinite loop gain and other application considerations.

Acceleration Feedforward

Acceleration Feedforward Gain scales the current Command Acceleration by the Acceleration Feedforward Gain and adds it as an offset to the Servo Output generated by the servo loop. With this done, the servo loops do not contribute much to the Servo Output and the Position and/or Velocity Error values are significantly reduced. Therefore, when used in conjunction with the Velocity Feedforward Gain, the Acceleration Feedforward Gain allows the following error of the servo system during the acceleration and deceleration phases of motion to be reduced to nearly zero. This is important in applications such as electronic gearing, position camming, and synchronization applications, where it is necessary that the actual axis position not significantly lag behind the commanded position at any time. The optimal value for Acceleration Feedforward is 100%, theoretically. In reality, however, you may need to update the value to accommodate velocity loops with non-infinite loop gain and other application considerations.

Proportional (Position) Gain

Position Error is multiplied by the Position Loop Proportional Gain, or Pos P Gain, to produce a component to the Velocity Command that ultimately attempts to correct for the position error. Too little Pos P Gain results in excessively compliant, or mushy, axis behavior. Too large a Pos P Gain, on the other hand, can result in axis oscillation due to classical servo instability.

To set the gain manually, you must first set the Torque scaling in the **Output** tab of this dialog box.

If you know the desired loop gain in inches per minute per mil or millimeters per minute per mil, use the following formula to calculate the corresponding P gain:

Pos P Gain = 16.667 * Desired Loop Gain (IPM/mil)

If you know the desired unity gain bandwidth of the position servo in Hertz, use the following formula to calculate the corresponding P gain:

Pos P Gain = Bandwidth (Hertz) * 6.28

The typical value for the Position Proportional Gain is ~100 Sec-1.

Integral (Position) Gain

The Integral (that is, summation) of Position Error is multiplied by the Position Loop Integral Gain, or Pos I Gain, to produce a component to the Velocity

Command that ultimately attempts to correct for the position error. Pos I Gain improves the steady-state positioning performance of the system. Increasing the integral gain generally increases the ultimate positioning accuracy of the system. Excessive integral gain, however, results in system instability.

In certain cases, Pos I Gain control is disabled. One such case is when the servo output to the axis drive is saturated. Continuing integral control behavior in this case would only exacerbate the situation. When the **Integrator Hold** parameter is set to **Enabled**, the servo loop automatically disables the integrator during commanded motion.

While the Pos I Gain, if employed, is typically established by the automatic servo tuning procedure (in the **Tuning** tab of this dialog box), the Pos I Gain value may also be set manually. However, it must be stressed that the Torque Scaling factor for the axis must be established for the drive system (in the **Output** tab of this dialog box). Once this is done, the Pos I Gain can be computed based on the current or computed value for the Pos P Gain using the following formula:

Pos I Gain = .025 * 0.001 Sec/mSec * (Pos P Gain)2

Assuming a Pos P Gain value of 100 Sec-1, this results in a Pos I Gain value of 2.5 ~0.1 mSec-1 - Sec-1.

Proportional (Velocity) Gain

This parameter is enabled only for external drives configured for Torque loop operation in the **Servo** tab.

Velocity Error is multiplied by the Velocity Proportional Gain to produce a component to the Torque Command that ultimately attempts to correct for the velocity error, creating a damping effect. Thus, increasing the Velocity Proportional Gain results in smoother motion, enhanced acceleration, reduced overshoot, and greater system stability. However, too much Velocity Proportional Gain leads to high frequency instability and resonance effects.

If you know the desired unity gain bandwidth of the velocity servo in Hertz, you can use the following formula to calculate the corresponding P gain.

Vel P Gain = Bandwidth (Hertz) / 6.28

The typical value for the Velocity Proportional Gain is ~250 mSec-1.

Integral (Velocity) Gain

This parameter is enabled only for external drives configured for Torque loop operation in the **Servo** tab.

At every servo update the current Velocity Error is accumulated in a variable called the Velocity Integral Error. This value is multiplied by the Velocity Integral Gain to produce a component to the Torque Command that attempts to correct for the velocity error. The higher the Vel I Gain value, the faster the axis is driven to the zero Velocity Error condition. Unfortunately, I Gain control is intrinsically unstable. Too much I Gain results in axis oscillation and servo instability.

In certain cases, Vel I Gain control is disabled. One such case is when the servo output to the axis' drive is saturated. Continuing integral control behavior in this case would only exacerbate the situation. When the **Integrator Hold** parameter is set to **Enabled**, the servo loop automatically disables the integrator during commanded motion.

Due to the destabilizing nature of Integral Gain, it is recommended that Position Integral Gain and Velocity Integral Gain be considered mutually exclusive. If Integral Gain is needed for the application, use one or the other. In general, where static positioning accuracy is required, Position Integral Gain is the better choice.

While the Vel I Gain, if employed, is typically established by the automatic servo tuning procedure (in the **Tune** tab of this dialog box), the Pos I Gain value may also be set manually. Before doing this, it must be stressed that the Torque Scaling factor for the axis must be established for the drive system in the **Output** tab. Once this is done, the Vel I Gain can be computed based on the current or computed value for the Vel P Gain using the following formula:

Vel I Gain = 0.25 * 0.001 Sec/mSec * (Vel P Gain)2

The typical value for the Velocity Proportional Gain is ~15 mSec-2.

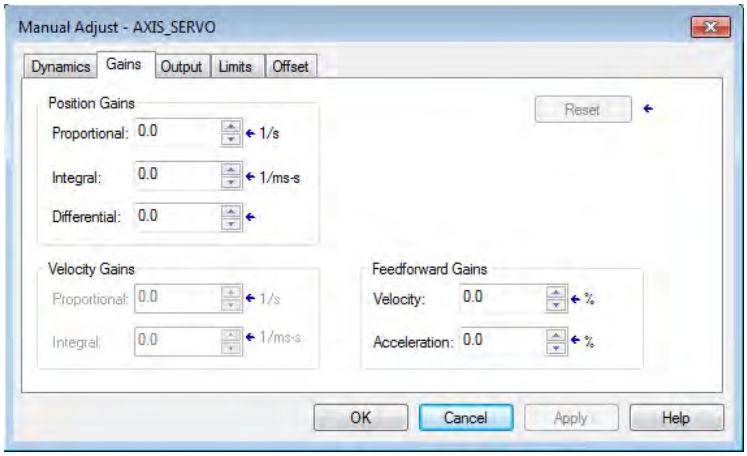
Integrator Hold

If the Integrator Hold parameter is set to:

- Enabled, the servo loop temporarily disables any enabled position or velocity integrators while the command position is changing. This feature is used by point-to-point moves to minimize the integrator wind-up during motion.
- Disabled, all active position or velocity integrators are enabled.

Manual Adjust

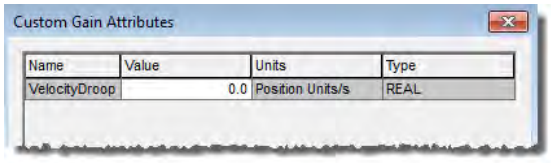
Opens the **Gains** tab of the **Manual Adjust** dialog box for online editing.



Manual Adjust is unavailable when Logix Designer application is in Wizard mode, and when you have not yet saved or applied your offline edits to the parameters.

Set custom gains

Use the **Custom Gain Attributes** dialog box to edit the **VelocityDroop** attribute.



When a parameter transitions to a read-only state, any pending changes to parameter values are lost, and the parameter reverts to the most recently saved parameter value.

Attribute

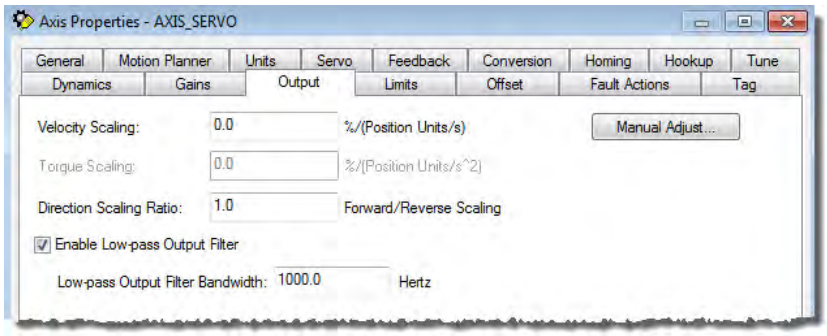
The following attribute value can be monitored and edited in this dialog box.

Attribute	Description
VelocityDroop	This 32-bit unsigned attribute – also referred to as static gain – acts as a very slow discharge of the velocity loop integrator. VelocityDroop may be used as a component of an external position loop system where setting this parameter to a higher, nonzero value eliminates servo hunting due to load/stick friction effects. This parameter only has effect if VelocityIntegralGain is not zero. Its value ranges from 0 to 2.14748x10^12. This value is not applicable for Ultra3000 drives.

Output tab - AXIS_SERVO

Use the **Output** tab for offline configuration for an axis of the type AXIS_SERVO that is configured in the **General** tab as a Servo drive. You can configure the following values.

- Set the torque scaling value, which is used to generate gains.
- Enable and configure the low-pass digital output filter.



The parameters on this tab can be edited in one of the following ways.

- Edit on this tab by typing your parameter changes and then click **OK** to save your edits.
- Edit in the Manual Adjust dialog box: Click **Manual Adjust** to open the **Manual Adjust** dialog box to this tab and use the spin controls to edit parameter settings. Your changes are saved the moment a spin control changes any parameter value.

The parameters on this tab become read-only and cannot be edited when the controller is online if the controller is set to Hard Run mode, or if a Feedback On condition exists.

When Logix Designer application is offline, the following parameters can be edited and the program saved to disk using the Save command or by clicking **Apply**. You must re-download the edited program to the controller before it can be run.

Velocity Scaling

The Velocity Scaling attribute is used to convert the output of the servo loop into equivalent voltage to an external velocity drive. This has the effect of 'normalizing' the units of the servo loop gain parameters so that their values are not affected by variations in feedback resolution, drive scaling, or mechanical gear ratios. The Velocity Scaling value is typically established by the servo's automatic tuning procedure, but these values can be calculated, if necessary, using the following guidelines.

If the axis is configured for a velocity external drive (in the **Servo** tab of this dialog box), the software velocity loop in the servo module is disabled. In this case, the Velocity Scaling value can be calculated by the following formula:

Velocity Scaling = 100% / (Speed @ 100%)

For example, if this axis is using position units of motor revolutions (revs), and the drive is scaled such that with an input of 100% (for example, 10 Volts) the motor goes 5,000 RPM (or 83.3 RPS), the Velocity Scaling attribute value would be calculated as:

Velocity Scaling = 100% / (83.3 RPS) = 1.2% / Revs Per Second

Torque/Force Scaling

The Torque Scaling attribute is used to convert the acceleration of the servo loop into equivalent % rated torque to the motor. This has the effect of 'normalizing' the units of the servo loops gain parameters so that their values are not affected by variations in feedback resolution, drive scaling, motor and load inertia, and mechanical gear ratios. The Torque Scaling value is typically established by the controller's automatic tuning procedure, but the value can be manually calculated, if necessary, using the following guidelines:

Torque Scaling = 100% Rated Torque / (Acceleration @ 100% Rated Torque)

For example, if this axis is using position units of motor revolutions (revs), with 100% rated torque applied to the motor, if the motor accelerates at a rate of 3000 Revs/Sec2, the Torque Scaling attribute value would be calculated as follows.

Torque Scaling = 100% Rated / (3000 RPS2) = 0.0333% Rated/ Revs Per Second2

If the Torque Scaling value does not reflect the true torque to acceleration characteristic of the system, the gains also do not reflect the true performance of the system.

Direction Scaling Ratio

The ratio between the extend direction gain and the retract direction gain.

Tip: This field is disabled for the 1756-M02AE module.

Enable Low Pass Output Filter

Select this to enable the servo's low-pass digital output filter. Clear this to disable this filter.

During tuning, if the controller detects a high degree of tuning inertia, it enables the Low Pass Output Filter and calculates and sets a value for Low Pass Output Filter Bandwidth.

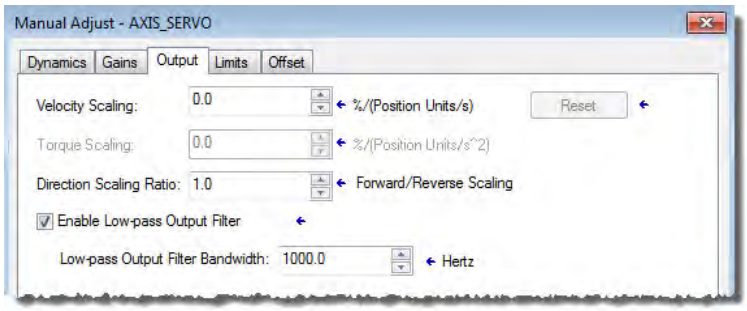
Low-pass Output Filter Bandwidth

When **Enable Low-pass Output Filter** is selected, this value sets the bandwidth, in Hertz, of the servo's low-pass digital output filter. Use this output filter to filter out high frequency variation of the servo module output to the drive. All output from the servo module greater than the Filter Bandwidth setting is filtered and not sent to the drive.

If the Low-pass Output Filter Bandwidth value is set to zero, the low-pass output filter is disabled. The lower the Filter Bandwidth value, the greater the attenuation of these high frequency components of the output signal. Because the low-pass filter adds lag to the servo loop, which pushes the system towards instability, decreasing the Filter Bandwidth value usually requires lowering the Position or Velocity Proportional Gain settings to maintain stability. The output filter is particularly useful in high inertia applications where resonance behavior can severely restrict the maximum bandwidth capability of the servo loop.

Manual Adjust

Opens the **Output** tab of the **Manual Adjust** dialog box for online editing.

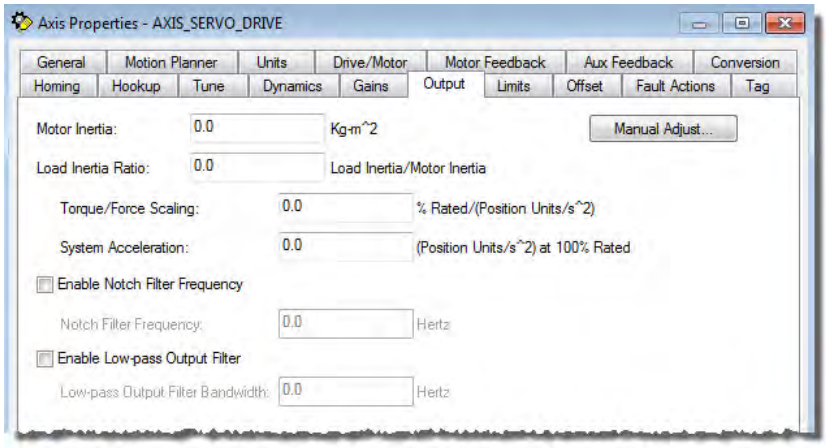


Manual Adjust is unavailable when Logix Designer application is in Wizard mode, and when you have not yet saved or applied your offline edits to the parameters.

Output tab - AXIS_SERVO_DRIVE

Use the **Output** tab for offline configuration for an axis of the type AXIS_SERVO_DRIVE, that is configured in the **General** tab as a Servo drive. You can configure the following values.

- Set the torque scaling value, which is used to generate gains.
- Enable and configure the Notch Filter.
- Enable and configure the low-pass digital output filter.



The parameters on this tab can be edited in the following ways.

- Edit on this tab by typing your parameter changes and then click **OK** to save your edits.
- Edit in the Manual Adjust dialog box: Click **Manual Adjust** to open the **Manual Adjust** dialog box to this tab and use the spin controls to edit

parameter settings. Your changes are saved the moment a spin control changes any parameter value.

The parameters on this tab become read-only and cannot be edited when the controller is online if the controller is set to Hard Run mode, or if a Feedback On condition exists.

When Logix Designer application is offline, the following parameters can be edited and the program saved to disk using the Save command or by clicking **Apply**. You must re-download the edited program to the controller before it can be run.

Motor Inertia

The Motor Inertia value represents the inertia of the motor without any load attached to the motor shaft in Torque Scaling units.

Load Inertia Ratio

The Load Inertia Ratio value represents the ratio of the load inertia to the motor inertia.

Torque/Force Scaling

The Torque Scaling attribute is used to convert the acceleration of the servo loop into equivalent % rated torque to the motor. This has the effect of normalizing the units of the servo loops gain parameters so that their values are not affected by variations in feedback resolution, drive scaling, motor and load inertia, and mechanical gear ratios. The Torque Scaling value is typically established by the controller's automatic tuning procedure, but the value can be manually calculated, if necessary, using the following guidelines:

Torque Scaling = 100% Rated Torque / (Acceleration @ 100% Rated Torque)

For example, if this axis is using position units of motor revolutions (revs), with 100% rated torque applied to the motor, if the motor accelerates at a rate of 3000 Revs/Sec2, the Torque Scaling attribute value would be calculated as follows.

Torque Scaling = 100% Rated / (3000 RPS2) = 0.0333% Rated/ Revs Per Second2

If the Torque Scaling value does not reflect the true torque to acceleration characteristic of the system, the gains also do not reflect the true performance of the system.

Enable Notch Filter Frequency

Select this to enable the drive's notch filter. Clear this to disable this filter.

Notch Filter Frequency

When **Enable Notch Filter Frequency** is selected, this value sets the center frequency of the drive's digital notch filter. If the Notch Filter value is set to zero, the notch filter Frequency is disabled.

Currently implemented as a 2nd order digital filter with a fixed Q, the Notch Filter provides approximately 40DB of output attenuation at the Notch Filter frequency. This output notch filter is particularly useful in attenuating mechanical resonance phenomena. The output filter is particularly useful in high inertia applications where mechanical resonance behavior can severely restrict the maximum bandwidth capability of the servo loop.

This value is not applicable for Ultra3000 drives.

Enable Low Pass Output Filter

Select this to enable the servo's low-pass digital output filter. Clear this to disable this filter.

During tuning, if the controller detects a high degree of tuning inertia, the controller enables the Low Pass Output Filter and calculates and sets a value for Low Pass Output Filter Bandwidth.

Low-pass Output Filter Bandwidth

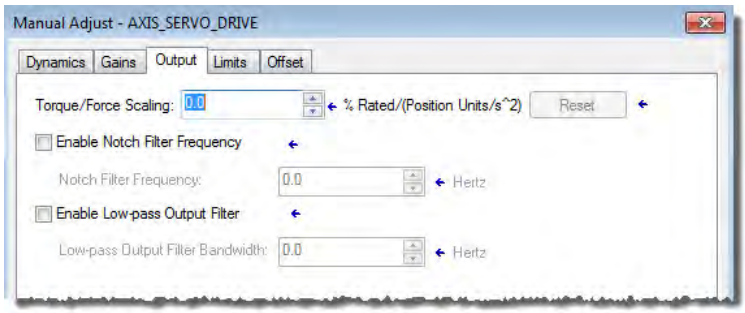
When **Enable Low-pass Output Filter** is selected, this value sets the bandwidth, in Hertz, of the servo's low-pass digital output filter. Use this output filter to filter out high frequency variation of the servo module output to the drive. All output from the servo module greater than the Filter Bandwidth setting is filtered and not sent to the drive.

If the Low-pass Output Filter Bandwidth value is set to zero, the low-pass output filter is disabled. The lower the Filter Bandwidth value, the greater the attenuation

of these high frequency components of the output signal. Because the low-pass filter adds lag to the servo loop, which pushes the system towards instability, decreasing the Filter Bandwidth value usually requires lowering the Position or Velocity Proportional Gain settings to maintain stability. The output filter is particularly useful in high inertia applications where resonance behavior can severely restrict the maximum bandwidth capability of the servo loop.

Manual Adjust

Opens the **Output** tab of the **Manual Adjust** dialog box for online editing of **Torque/Force Scaling**, the **Notch Filter Frequency**, and the **Low-pass Output Filter** parameters.

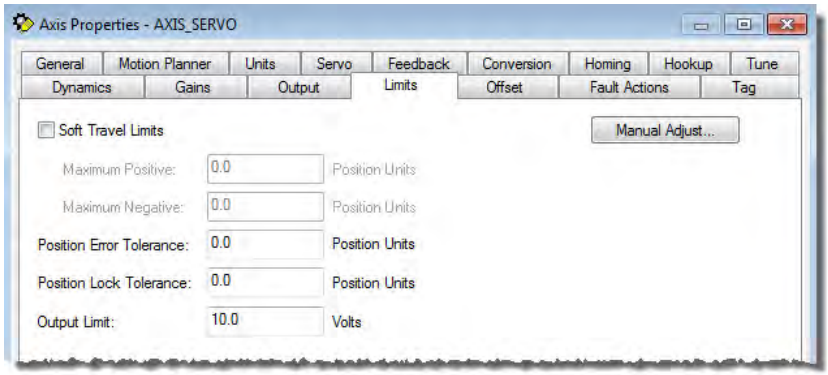


Manual Adjust is unavailable when Logix Designer application is in Wizard mode and when offline edits to the parameters have not yet been applied.

Limits tab - AXIS_SERVO

Use the **Limits** tab for offline configuration for an axis of the type AXIS_SERVO configured in the **General** tab as a Servo drive. You can configure the following values.

- Enable and set maximum positive and negative software travel limits.
- Configure Position Error Tolerance and Position Lock Tolerance.
- Set the drive's Output Limit.



The parameters on this tab can be edited in the following ways.

- Edit on this tab by typing your parameter changes and then clicking **OK** to save your edits.
- Edit in the Manual Adjust dialog box: click **Manual Adjust** to open the **Manual Adjust** dialog box to this tab and use the spin controls to edit parameter settings. Your changes are saved the moment a spin control changes any parameter value.

The parameters on this tab become read-only and cannot be edited when the controller is online if the controller is set to Run mode, or if a Feedback On condition exists.

When Logix Designer application is offline, the following parameters can be edited and the program saved or applied. You must download the edited program to the controller before it can be run.

Soft Travel Limits

Enables software overtravel checking for an axis when **Positioning Mode** is set to **Linear** (in the **Conversion** tab of this dialog box). If an axis is configured for software overtravel limits and if that axis passes beyond these maximum travel limits (positive or negative), a software overtravel fault is issued. The response to this fault is specified by the **Soft Overtravel** setting (in the **Fault Actions** tab of this dialog box). Software overtravel limits are disabled during the tuning process.

Maximum Positive

Type the maximum positive position to be used for software overtravel checking, in position units.

The Maximum Positive limit must be greater than the Maximum Negative limit.

Maximum Negative

Type the maximum negative position to be used for software overtravel checking, in position units.

The Maximum Negative limit must be less than the Maximum Positive limit.

Position Error Tolerance

Specifies how much position error the servo tolerates before issuing a position error fault. This value is interpreted as a +/- quantity.

For example, setting **Position Error Tolerance** to 0.75 position units means that a position error fault is generated whenever the position error of the axis is greater than 0.75 or less than -0.75 position units.

This value is set to twice the following error at maximum speed based on the measured response of the axis during the autotuning process. In most applications, this value provides reasonable protection in case of an axis fault or stall condition without nuisance faults during normal operation. If you must change the calculated position error tolerance value, the recommended setting is 150% to 200% of the position error while the axis is running at its maximum speed.

Position Lock Tolerance

Specifies the maximum position error the servo module accepts to indicate the Position Lock status bit is set. This is useful in determining when the desired end position is reached for position moves. This value is interpreted as a +/- quantity.

For example, specifying a lock tolerance of 0.01 provides a minimum positioning accuracy of +/- 0.01 position units.

Output limit

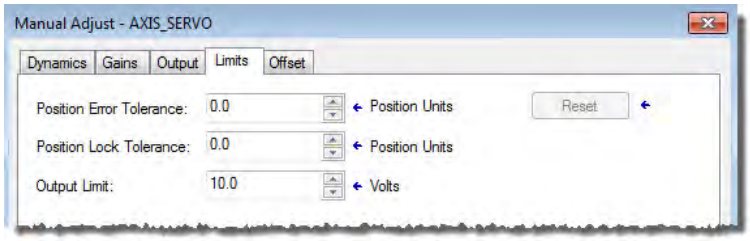
Provides a method of limiting the maximum servo output voltage of a physical axis to a specified level. The servo output for the axis as a function of position servo error, with and without servo output limiting, is shown below.

The servo output limit may be used as a software current or torque limit if you are using a drive in torque loop mode. The percentage of the drive's maximum current that the servo controller ever commands is equal to the specified servo output limit. For example, if the drive is capable of 30 Amps of current for a 10 Volt input, setting the servo output limit to 5V limits the maximum drive current to 15 Amps.

The servo output limit may also be used if the drive cannot accept the full ±10 Volt range of the servo output. In this case, the servo output limit value effectively limits the maximum command sent to the amplifier. For example, if the drive can only accept command signals up to ±7.5 Volts, set the servo output limit value to 7.5 volts.

Manual Adjust

Opens the **Limits** tab of the **Manual Adjust** dialog box for online editing of the **Position Error Tolerance**, **Position Lock Tolerance**, and **Output Limit** parameters.

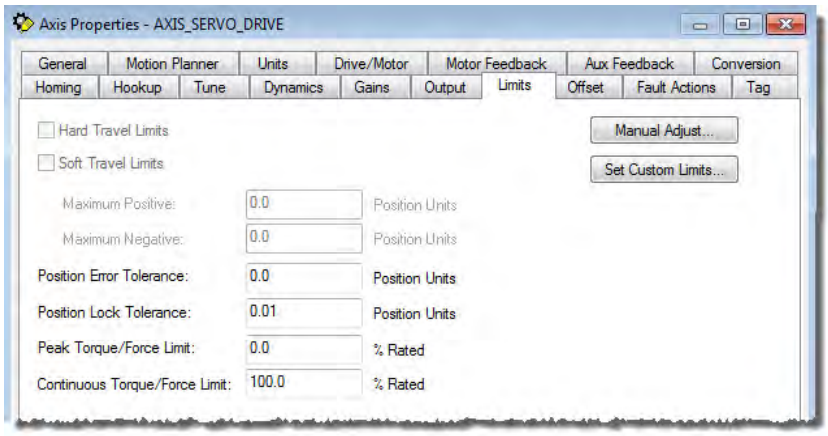


Manual Adjust is disabled when Logix Designer application is in Wizard mode, and when offline edits to the parameters have not yet been saved or applied.

Limits tab - AXIS_SERVO_DRIVE

Use the **Limits** tab to make the following offline configurations for an axis of the type AXIS_SERVO_DRIVE configured as a Servo drive in the **General** tab of this dialog box.

- Enable and set maximum positive and negative software travel limits.
- Configure Position Error Tolerance and Position Lock Tolerance.



The parameters on this tab can be edited in the following ways.

- Edit on this tab by typing your parameter changes and then click **OK** to save your edits.
- Edit in the Manual Adjust dialog box: click **Manual Adjust** to open the **Manual Adjust** dialog box to this tab and use the spin controls to edit parameter settings. Your changes are saved the moment a spin control changes any parameter value.

The parameters on this tab become read-only and cannot be edited when the controller is online if the controller is set to Hard Run mode, or if a Feedback On condition exists.

When Logix Designer application is offline, the following parameters can be edited and the program saved to disk using the Save command or by clicking **Apply**. You must re-download the edited program to the controller before it can be run.

Hard Travel Limits

Enables a periodic test that monitors the current state of the positive and negative overtravel limit switch inputs, when **Positioning Mode** is set to **Linear** (in the **Conversion** tab of this dialog box). If an axis is configured for hardware overtravel checking and if that axis passes beyond a positive or negative overtravel limit switch, a Positive Hard Overtravel Fault or Negative Hard Overtravel Fault is issued. The response to this fault is specified by the **Hard Overtravel** setting (in the **Fault Actions** tab of this dialog box).

Soft Travel Limits

Enables software overtravel checking for an axis when **Positioning Mode** is set to **Linear** (in the **Conversion tab** of this dialog box). If an axis is configured for software overtravel limits and if that axis passes beyond these maximum travel limits (positive or negative), a software overtravel fault is issued. The response to this fault is specified by the **Soft Overtravel** setting (in the **Fault Actions** tab of this dialog box). Software overtravel limits are disabled during the tuning process.

Maximum Positive

Type the maximum positive position to be used for software overtravel checking, in position units.

The Maximum Positive limit must be greater than the Maximum Negative limit.

Maximum Negative

Type the maximum negative position to be used for software overtravel checking, in position units.

The Maximum Negative limit must be less than the Maximum Positive limit.

Position Error Tolerance

Specifies how much position error the servo tolerates before issuing a position error fault. This value is interpreted as a +/- quantity.

For example, setting **Position Error Tolerance** to 0.75 position units means that a position error fault is generated whenever the position error of the axis is greater than 0.75 or less than -0.75 position units.

This value is set to twice the following error at maximum speed based on the measured response of the axis, during the autotuning process. In most applications, this value provides reasonable protection in case of an axis fault or stall condition without nuisance faults during normal operation. If you must change the calculated position error tolerance value, the recommended setting is 150% to 200% of the position error while the axis is running at its maximum speed.

Position Lock Tolerance

Specifies the maximum position error the servo module accepts to indicate the Position Lock status bit is set. This is useful in determining when the desired end position is reached for position moves. This value is interpreted as a +/- quantity.

For example, specifying a lock tolerance of 0.01 provides a minimum positioning accuracy of +/- 0.01 position units.

Peak Torque/Force Limit

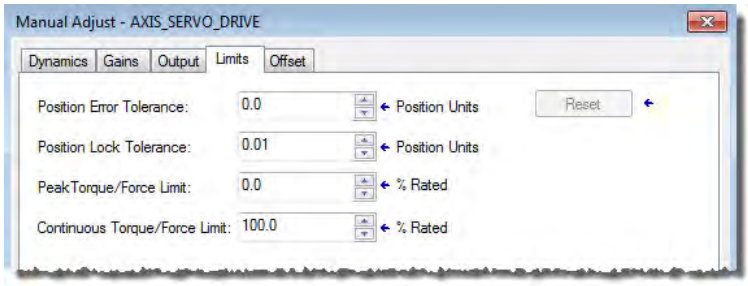
The Peak Torque/Force Limit specifies the maximum percentage of the motors rated current that the drive can command as positive or negative torque/force. For example, a torque limit of 150% shall limit the current delivered to the motor to 1.5 times the continuous current rating of the motor.

Continuous Torque/Force Limit

The Continuous Torque/Force Limit specifies the maximum percentage of the motors rated current that the drive can command on a continuous or RMS basis. For example, a Continuous Torque/Force Limit of 150% limits the continuous current delivered to the motor to 1.5 times the continuous current rating of the motor.

Manual Adjust

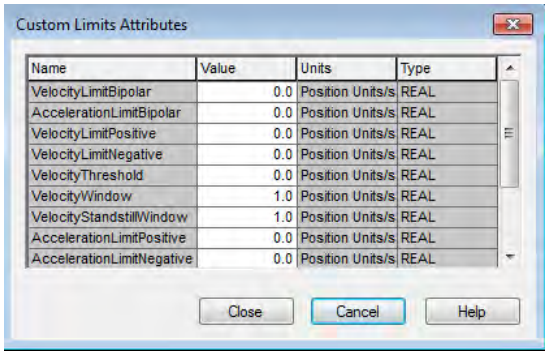
Opens the **Limits** tab of the **Manual Adjust** dialog box for online editing of the **Position Error Tolerance**, **Position Lock Tolerance**, **Peak Torque/Force Limit**, and **Continuous Torque/Force Limit** parameters.



Manual Adjust is unavailable when Logix Designer application is in Wizard mode, and when offline edits to the parameters have not yet been saved or applied.

Set custom limits

Opens the **Custom Limit Attributes** dialog box.



From this dialog box you can monitor and edit the limit-related attributes.

When Logix Designer application is online, the parameters on this tab transition to a read-only state. When a parameter transitions to a read-only state, any pending changes to parameter values are lost, and the parameter reverts to the most recently saved parameter value.

When multiple workstations connect to the same controller using Logix Designer application and invoke the **Axis Wizard** or **Axis Properties** dialog box, the firmware allows only the first workstation to make any changes to axis attributes. The second workstation switches to a Read Only mode, indicated in the title bar, so that you may view the changes from that workstation, but not edit them.

Attributes

The following attribute values can be monitored and edited in this dialog box.

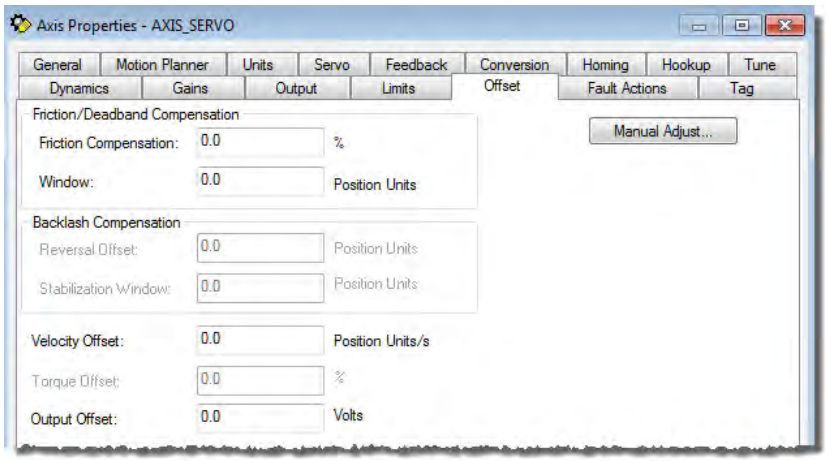
Attribute	Description
VelocityLimitBipolar	This attribute sets the velocity limit symmetrically in both directions. If the command velocity exceeds this value, VelocityLimitStatusBit of the DriveStatus attribute is set. This attribute has a value range of 0 to 2.14748x1012.
AccelerationLimitBipolar	This attribute sets the acceleration and deceleration limits for the drive. If the command acceleration exceeds this value, AccelLimitStatusBit of the DriveStatus attribute is set. This attribute has a value range of 0 to 2.14748x1015.
TorqueLimitBipolar	This attribute sets the torque limit symmetrically in both directions. When actual torque exceeds this value, TorqueLimitStatus of the DriveStatus attribute is set. This attribute has a value range of 0 to 1000.
VelocityLimitPositive	This attribute displays the maximum allowable velocity in the positive direction. If the velocity limit is exceeded, bit 5 ('Velocity Command Above Velocity Limit') VelocityLimitStatusBit of the DriveStatus attribute is set. This attribute has a value range of 0 to 2.14748x1012.
VelocityLimitNegative	This attribute displays the maximum allowable velocity in the negative direction. If the velocity limit is exceeded, bit 5 ('Velocity Command Above Velocity Limit') VelocityLimitStatusBit of the DriveStatus attribute is set. This attribute has a value range of -2.14748x1012 to 0.
VelocityThreshold	This attribute displays the velocity threshold limit. If the motor velocity is less than this limit, VelocityThresholdStatus of the DriveStatus attribute is set. This attribute has a value range of 0 to 2.14748x1012.
VelocityWindow	This attribute displays the limits of the velocity window. If the motor's actual velocity differs from the command velocity by an amount less that this limit, VelocityLockStatus of the DriveStatus attribute is set. This attribute has a value range of 0 to 2.14748x1012.
VelocityStandstillWindow	This attribute displays the velocity limit for the standstill window. If the motor velocity is less than this limit VelocityStandStillStatus of the DriveStatus bit is set. This attribute has a value range of 0 to 2.14748x1012.

Attribute	Description
AccelerationLimitPositive	This attribute limits the maximum acceleration ability of the drive to the programmed value. If the command acceleration exceeds this value, AccelLimitStatusBit of the DriveStatus attribute is set. This attribute has a value range of 0 to 2.14748x1015.
AccelerationLimitNegative	This attribute limits the maximum acceleration ability of the drive to the programmed value. If the command acceleration exceeds this value, the AccelLimitStatus bit of the DriveStatus attribute is set. This attribute has a value range of -2.14748x1015 to 0.
TorqueLimitPositive	This attribute displays the maximum torque in the positive direction. If the torque limit is exceeded, the TorqueLimitStatus bit of the DriveStatus attribute is set. This attribute has a value range of 0 to 1000.
TorqueLimitNegative	This attribute displays the maximum torque in the negative direction. If the torque limit is exceeded, the TorqueLimitStatus bit of the DriveStatus attribute is set. This attribute has a value range of -1000 to 0.
TorqueThreshold	This attribute displays the torque threshold. If this limit is exceeded, the TorqueThreshold bit of the DriveStatus attribute is set. This attribute has a value range of 0 to 1000.

Offset tab - AXIS_SERVO

Use the **Offset** tab to make offline adjustments to the following Servo Output values for an axis of the type AXIS_SERVO configured as a Servo drive in the **General** tab of this dialog box.

- **Backlash Compensation**
- **Velocity Offset**
- **Torque Offset**
- **Output Offset**



The parameters on this tab can be edited in the following ways.

- Edit on this tab by typing your parameter changes and then click **OK** to save your edits.
- Edit in the Manual Adjust dialog box: Click **Manual Adjust** to open the **Manual Adjust** dialog box to this tab and use the spin controls to edit parameter settings. Your changes are saved the moment a spin control changes any parameter value.

The parameters on this tab become read-only and cannot be edited when the controller is online if the controller is set to Hard Run mode, or if a Feedback On condition exists.

When Logix Designer application is offline, the following parameters can be edited and the program saved to disk using the Save command or by clicking **Apply**. You must re-download the edited program to the controller before it can be run.

Friction/Deadband Compensation and Backlash Compensation

The percentage of output level added to a positive current Servo Output value, or subtracted from a negative current Servo Output value, for the purpose of moving an axis that is stuck in place due to static friction.

It is not unusual for an axis to have enough static friction (called 'sticktion') that, even with a significant position error, the axis refuses to budge. Backlash Compensation is used to break 'sticktion' in the presence of a nonzero position error. This is done by adding, or subtracting, a percentage output level), called Backlash Compensation to the Servo Output value.

The Backlash Compensation value should be just less than the value that would break the 'sticktion'. A larger value can cause the axis to 'dither', that is, move rapidly back and forth about the commanded position.

This controller attribute is replicated in the motion module.

Backlash Compensation Window

To address the issue of dither when applying Backlash Compensation and hunting from the integral gain, a Backlash Compensation Window is applied around the current command position when the axis is not being commanded to move. If the actual position is within the Backlash Compensation Window, the Backlash Compensation value is applied to the Servo Output, but scaled by the ratio of the position error to the Backlash Compensation Window. Within the window, the servo integrators are also disabled. Thus, once the position error reaches or exceeds the value of the Backlash Compensation Window attribute, the full Backlash Compensation value is applied. If the Backlash Compensation Window is set to zero, this feature is effectively disabled.

A nonzero Backlash Compensation Window has the effect of softening the Backlash Compensation as it is applied to the Servo Output and reducing the dithering effect that it can create. This generally allows higher values of Backlash Compensation to be applied. Hunting is also eliminated at the cost of a small steady-state error.

Backlash Compensation and Backlash Reversal Offset

Backlash Reversal Offset provides the capability to compensate for positional inaccuracy introduced by mechanical backlash. For example, power-train type applications require a high level of accuracy and repeatability during machining operations. Axis motion is often generated by a number of mechanical components, a motor, a gearbox, and a ball-screw that may introduce inaccuracies and that are subject to wear over their lifetime. Therefore, when an axis is commanded to reverse direction, mechanical play in the machine (through the gearing, ball-screw, and so on) may result in a small amount of motor motion without axis motion. As a result, the feedback device may indicate movement even though the axis has not physically moved.

If a value of zero is applied to the Backlash Reversal Offset, the feature is effectively disabled. When it is enabled by a nonzero value, and the load is engaged by a reversal of the commanded motion, changing the Backlash Reversal Offset can cause the axis to shift as the offset correction is applied to the command position.

Stabilization Window

The Backlash Stabilization Window controls the Backlash Stabilization feature in the servo control loop.

Configuring a suitable value for the Backlash Stabilization Window eliminates gearbox buzz without sacrificing any servo performance. In general, this value should be set to the measured backlash distance. A Backlash Stabilization Window value of zero effectively disables the feature.

Velocity Offset

Provides a dynamic velocity correction to the output of the position servo loop, in position units per second.

Torque Offset

Provides a dynamic torque command correction to the output of the velocity servo loop, as a percentage of velocity servo loop output.

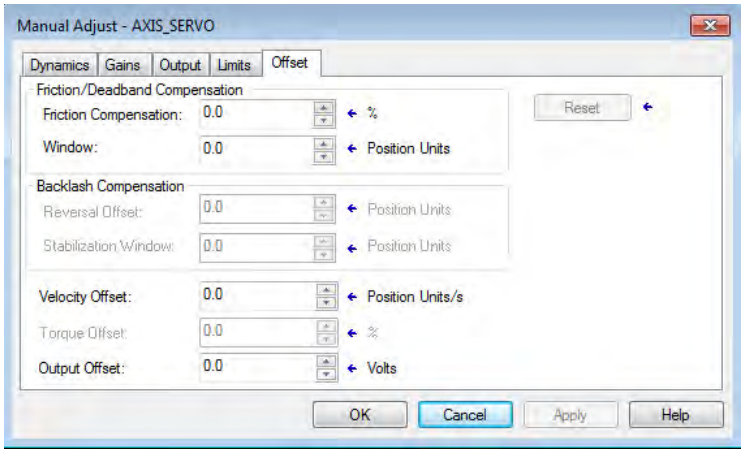
Output Offset

Corrects the problem of axis 'drift' by adding a fixed voltage value (not to exceed ±10 Volts) to the Servo Output value. Enter a value to achieve near zero drive velocity when the uncompensated Servo Output value is zero.

When interfacing an external drive, it is necessary to compensate for the effect of drive offset. This is especially true for a velocity drive. Cumulative offsets of the servo module's DAC output and the Servo Drive Input result in a situation where a zero commanded Servo Output value causes the axis to 'drift'. If the drift is excessive, it can cause problems with the Hookup Diagnostic and Tuning procedures, and can result in a steady-state nonzero position error when the servo loop is closed.

Manual Adjust

Opens the **Offset** tab of the **Manual Adjust** dialog box for online editing of the **Friction/Deadband Compensation**, **Backlash Compensation**, **Velocity Offset**, **Torque Offset**, and **Output Offset** parameters.

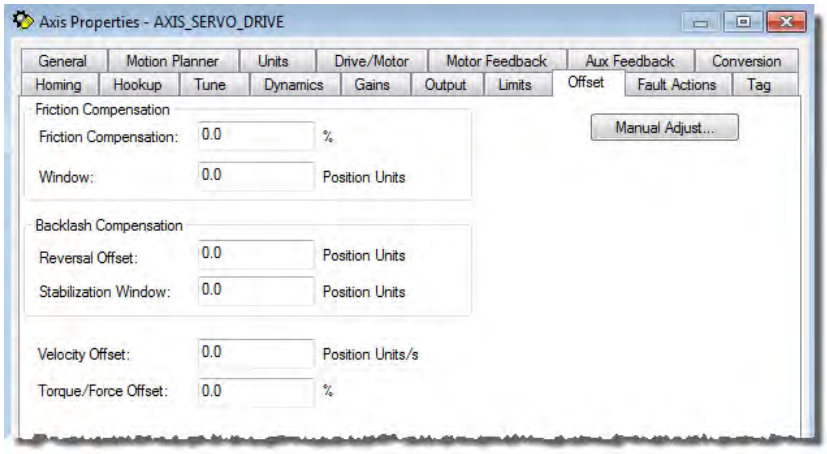


Manual Adjust is unavailable when Logix Designer application is in Wizard mode, and when offline edits to the parameters have not yet been saved or applied.

Offset tab - AXIS_SERVO_DRIVE

Use the **Offset** tab to make offline adjustments to the following Servo Output values for an axis of the type AXIS_SERVO_DRIVE configured as a Servo drive in the **General** tab of this dialog box.

- **Backlash Compensation**
- **Velocity Offset**
- **Torque/Force Offset**



The parameters on this tab can be edited in the following ways.

- Edit on this tab by typing your parameter changes and then click **OK** to save your edits
- Edit in the **Manual Adjust** dialog box: Click **Manual Adjust** to open the **Manual Adjust** dialog box to this tab and use the spin controls to edit parameter settings. Your changes are saved the moment a spin control changes any parameter value.

The parameters on this tab become read-only and cannot be edited when the controller is online if the controller is set to Hard Run mode, or if a Feedback On condition exists.

When Logix Designer application is offline, the following parameters can be edited and the program saved to disk using the Save command or by clicking **Apply**. You must re-download the edited program to the controller before it can be run.

Backlash Compensation

The percentage of output level added to a positive current Servo Output value, or subtracted from a negative current Servo Output value, for the purpose of moving an axis that is stuck in place due to static friction.

It is not unusual for an axis to have enough static friction – called 'sticktion' – that, even with a significant position error, the axis refuses to budge. Backlash Compensation is used to break 'sticktion' in the presence of a nonzero position error. This is done by adding, or subtracting, a percentage output level, called Backlash Compensation to the Servo Output value.

The Backlash Compensation value should be just less than the value that would break the ‘sticktion’. A larger value can cause the axis to ‘dither’, that is, move rapidly back and forth about the commanded position.

Backlash Compensation Window

To address the issue of dither when applying Backlash Compensation and hunting from the integral gain, a Backlash Compensation Window is applied around the current command position when the axis is not being commanded to move. If the actual position is within the Backlash Compensation Window, the Backlash Compensation value is applied to the Servo Output, but scaled by the ratio of the position error to the Backlash Compensation Window. Within the window, the servo integrators are also disabled. Thus, once the position error reaches or exceeds the value of the Backlash Compensation Window attribute, the full Backlash Compensation value is applied. If the Backlash Compensation Window is set to zero, this feature is effectively disabled.

A nonzero Backlash Compensation Window has the effect of softening the Backlash Compensation as it is applied to the Servo Output and reducing the dithering effect that it can create. This generally allows higher values of Backlash Compensation to be applied. Hunting is also eliminated at the cost of a small steady-state error.

Backlash Compensation and Backlash Reversal Offset

Backlash Reversal Offset provides the capability to compensate for positional inaccuracy introduced by mechanical backlash. For example, power-train type applications require a high level of accuracy and repeatability during machining operations. Axis motion is often generated by a number of mechanical components, a motor, a gearbox, and a ball-screw that may introduce inaccuracies and that are subject to wear over their lifetime. Therefore, when an axis is commanded to reverse direction, mechanical play in the machine (through the gearing, ball-screw, and so on) may result in a small amount of motor motion without axis motion. As a result, the feedback device may indicate movement even though the axis has not physically moved.

If a value of zero is applied to the Backlash Reversal Offset, the feature is unavailable. When it is enabled by a nonzero value, and the load is engaged by a reversal of the commanded motion, changing the Backlash Reversal Offset can cause the axis to shift as the offset correction is applied to the command position.

Stabilization Window

The Backlash Stabilization Window controls the Backlash Stabilization feature in the servo control loop.

Configuring a suitable value for the Backlash Stabilization Window eliminates gearbox buzz without sacrificing any servo performance. In general, this value should be set to the measured backlash distance. This feature is unavailable when a Backlash Stabilization Window value is zero.

Velocity Offset

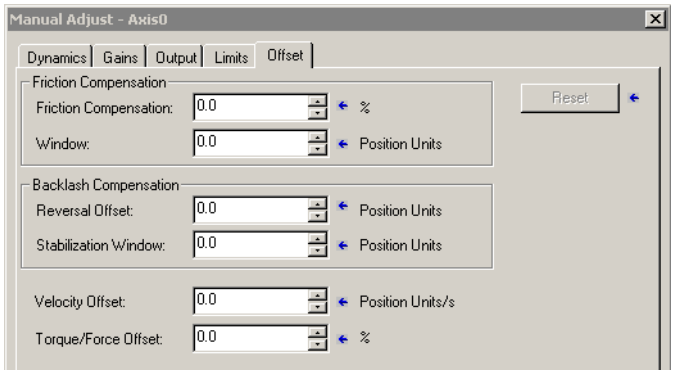
Provides a dynamic velocity correction to the output of the position servo loop, in position units per second.

Torque/Force Offset

Provides a dynamic torque command correction to the output of the velocity servo loop, as a percentage of velocity servo loop output.

Manual adjust for Offset tab

Opens the **Offset** tab of the Manual Adjust dialog box for online editing of the **Friction/Deadband Compensation**, **Backlash Compensation**, **Velocity Offset**, **Torque Offset**, and **Output Offset** parameters.



Manual Adjust is unavailable when Logix Designer application is in Wizard mode, and when offline edits to the parameters have not yet been saved or applied.

Fault Actions tab - AXIS_SERVO

Use the **Fault Actions** tab to specify the actions that are taken in response to the following faults.

- Drive Fault
- Feedback Noise Fault
- Feedback Loss Fault
- Position Error Fault

- Soft Overtravel Fault

When a parameter transitions to a read-only state, any pending changes to parameter values are lost, and the parameter reverts to the most recently saved parameter value.

When multiple workstations connect to the same controller using Logix Designer application and invoke the **Axis Wizard** or **Axis Properties** dialog box, the firmware allows only the first workstation to make any changes to axis attributes. The second workstation switches to a Read Only mode, indicated in the title bar, so that you may view the changes from that workstation, but not edit them.

Select one of the following fault actions for each fault type:

- Shutdown - If a fault action is set to Shutdown, then when the associated fault occurs, axis servo action is unavailable, the servo amplifier output is zeroed, and the drive enable output is deactivated. Shutdown is the most severe action to a fault and it is usually reserved for faults that could endanger the machine or the operator if power is not removed as quickly and completely as possible.
- Disable Drive - If a fault action is set to Disable Drive, then when the associated fault occurs, axis servo action is unavailable, the servo amplifier output is zeroed, and the drive enable output is deactivated.
- Stop Motion - If a fault action is set to Stop Motion, then when the associated fault occurs, the axis immediately starts decelerating the axis command position to a stop at the configured Maximum Deceleration Rate without disabling servo action or the servo modules Drive Enable output. This is the gentlest stopping mechanism in response to a fault. It is usually used for less severe faults. After the stop command fault action has stopped the axis, no further motion can be generated until the fault is first cleared.
- Status Only - If a fault action is set to Status Only, then when the associated fault occurs, no action is taken. The application program must handle any

motion faults. In general, this setting should only be used in applications where the standard fault actions are not appropriate.

ATTENTION: Selecting the wrong fault action for your application can cause a dangerous condition resulting in unexpected motion, damage to the equipment, and physical injury or death. Keep clear of moving machinery.

Drive Fault

Use the Drive Fault field to specify the fault action to be taken when a drive fault condition is detected, for an axis with the Drive Fault Input enabled (in the **Servo** tab of this dialog box) that is configured as Servo (in the **General** tab of this dialog box). The available actions for this fault are **Shutdown**, **Disable Drive**, **Stop Motion**, and **Status Only**.

Feedback Noise

Use the Feedback Noise field to specify the fault action to be taken when excessive feedback noise is detected. The available actions for this fault are **Shutdown**, **Disable Drive**, **Stop Motion**, and **Status Only**.

Feedback Loss

Use the Feedback Loss field to specify the fault action to be taken when feedback loss condition is detected. The available actions for this fault are **Shutdown**, **Disable Drive**, **Stop Motion**, and **Status Only**.

Position Error

Use the **Position Error** box to specify the fault action to be taken when position error exceeds the position tolerance set for the axis, for an axis configured as Servo (in the **General** tab of this dialog box). The available actions for this fault are **Shutdown**, **Disable Drive**, **Stop Motion**, and **Status Only**.

Soft Overtravel

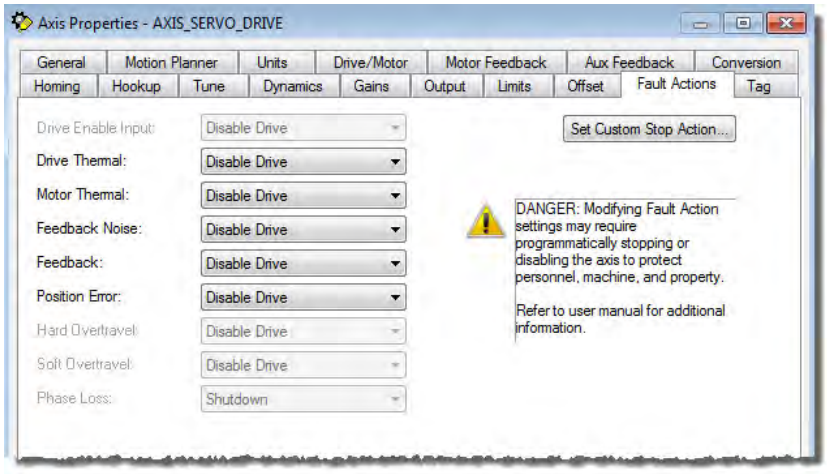
Use the Soft Overtravel field to specify the fault action to be taken when a software overtravel error occurs, for an axis with **Soft Travel Limits** enabled and configured (in the **Limits** tab of this dialog box) that is configured as **Servo** (in the

General tab of this dialog box). The available actions for this fault are **Shutdown**, **Disable Drive**, **Stop Motion**, and **Status Only**.

Fault Actions tab - AXIS_SERVO_DRIVE

Use the **Fault Actions** tab to specify the actions that are taken in response to the following faults for an axis of the type AXIS_SERVO_DRIVE.

- Drive Enable Input
- Drive Thermal Fault
- Motor Thermal Fault
- Feedback Noise Fault
- Feedback Fault
- Position Error Fault
- Hard Overtravel Fault
- Soft Overtravel Fault
- Phase Loss



When a parameter transitions to a read-only state, any pending changes to parameter values are lost, and the parameter reverts to the most recently saved parameter value.

When multiple workstations connect to the same controller using the Logix Designer application and invoke the **Axis Wizard** or **Axis Properties** dialog box, the firmware allows only the first workstation to make any changes to axis attributes. The second workstation switches to a Read Only mode, indicated in the title bar, so that you may view the changes from that workstation, but not edit them.

Select one of the following fault actions for each fault type.

- Shutdown - If a fault action is set to Shutdown, then when the associated fault occurs, axis servo action is immediately disabled, the servo amplifier output is zeroed, and the drive enable output is deactivated. Shutdown is the most severe action to a fault and it is usually reserved for faults that could endanger the machine or the operator if power is not removed as quickly and completely as possible.
- Disable Drive - If a fault action is set to Disable Drive, then when the associated fault occurs, it brings the axis to a stop by applying the Stopping Torque for up to the Stopping Time Limit. During this period, the servo is active but no longer tracking the command reference from Logix Designer. Once the axis is stopped (or the stopping limit is exceeded), the servo and power structure are disabled.
- Stop Motion - If a fault action is set to Stop Motion, then when the associated fault occurs, the axis immediately starts decelerating the axis command position to a stop at the configured Maximum Deceleration Rate without disabling servo action or the servo modules Drive Enable output. This is the gentlest stopping mechanism in response to a fault. It is usually used for less severe faults. After the stop command fault action has stopped the axis, no further motion can be generated until the fault is first cleared.
- Status Only - If a fault action is set to Status Only, then when the associated fault occurs, no action is taken. The application program must handle any motion faults. In general, this setting should only be used in applications where the standard fault actions are not appropriate.

ATTENTION: Selecting the wrong fault action for your application can cause a dangerous condition. Keep clear of moving machinery.

Drive Enable Input

Specifies the fault action to be taken when a Drive Enable Input Fault is detected, for an axis configured as **Servo** (in the **General** tab of this dialog box). The available actions for this fault are **Shutdown**, **Disable Drive**, **Stop Motion**, and **Status Only**.

Drive Thermal

Specifies the fault action to be taken when a Drive Thermal Fault is detected, for an axis configured as **Servo** (in the **General** tab of this dialog box). The available actions for this fault are **Shutdown**, **Disable Drive**, **Stop Motion**, and **Status Only**.

Motor Thermal

Specifies the fault action to be taken when a Motor Thermal Fault is detected, for an axis configured as Servo (in the **General** tab of this dialog box). The available actions for this fault are **Shutdown**, **Disable Drive**, **Stop Motion**, and **Status Only**.

Feedback Noise

Specifies the fault action to be taken when excessive feedback noise is detected. The available actions for this fault are **Shutdown**, **Disable Drive**, **Stop Motion**, and **Status Only**.

Feedback

Specifies the fault action to be taken when Feedback Fault is detected. The available actions for this fault are **Shutdown**, **Disable Drive**, **Stop Motion**, and **Status Only**.

Position Error

Specifies the fault action to be taken when position error exceeds the position tolerance set for the axis, for an axis configured as Servo (in the **General** tab of this dialog box). The available actions for this fault are **Shutdown**, **Disable Drive**, **Stop Motion**, and **Status Only**.

Hard Overtravel

Specifies the fault action to be taken when an axis encounters a travel limit switch, for an axis configured as Servo (in the **General** tab of this dialog box). The available actions for this fault are **Shutdown**, **Disable Drive**, **Stop Motion**, and **Status Only**.

Soft Overtravel

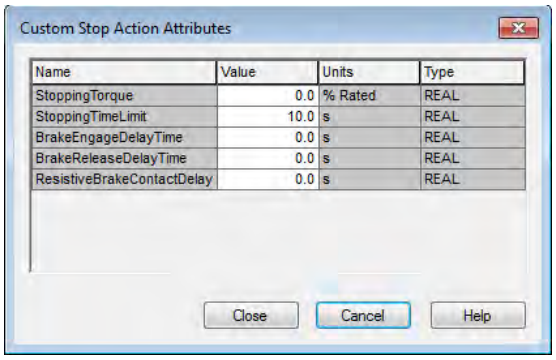
Specifies the fault action to be taken when a software overtravel error occurs, for an axis with **Soft Travel Limits** enabled and configured (in the **Limits** tab of this dialog box) that is configured as **Servo** (in the **General** tab of this dialog box). The available actions for this fault are **Shutdown**, **Disable Driv**e, **Stop Motion**, and **Status Only**.

Phase Loss

Specifies the fault action to be taken when a phase loss situation occurs for an axis configured as Servo (on the **General** tab of this dialog box). The available actions for this fault are **Shutdown**, **Disable Drive**, **Stop Motion** and **Status Only**. The default is **Shutdown**. When **Status Only** is chosen, Logix Designer motion commands continue and the drive uses available stored DC bus energy to operate the axes.

Set custom stop action

Opens the **Custom Stop Action Attributes** dialog box.



Use this dialog box to monitor and edit the Stop Action-related attributes.

When a parameter transitions to a read-only state, any pending changes to parameter values are lost, and the parameter reverts to the most recently saved parameter value.

When multiple workstations connect to the same controller using Logix Designer application and invoke the **Axis Wizard** or **Axis Properties** dialog box, the firmware allows only the first workstation to make any changes to axis attributes. The second workstation switches to a **Read Only** mode, indicated in the title bar, so that you may view the changes from that workstation, but not edit them.

Attributes

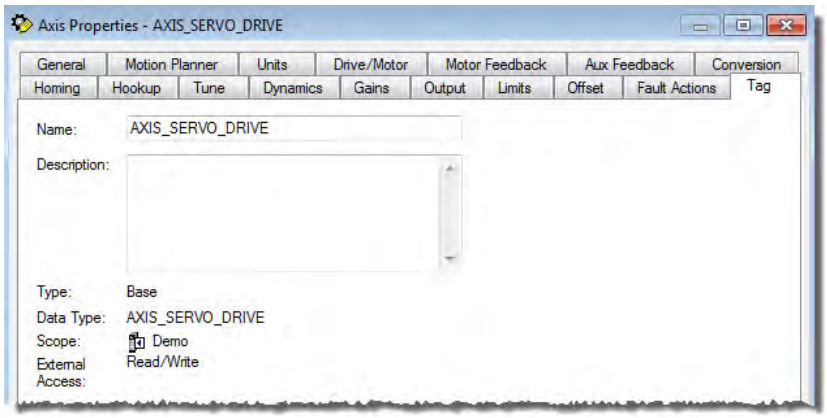
The following attribute, or parameter, values can be monitored and edited in this dialog box.

Attribute	Description
StoppingTorque	This attribute displays the amount of torque available to stop the motor. This attribute has a value range of 0...1000.

Attribute	Description
StoppingTimeLimit	This attribute displays the maximum amount of time that the drive amplifier remains enabled while trying to stop. It is useful for very slow velocity rate change settings. This attribute has a value range of 0...6553.5 units.
BrakeEngageDelayTime	When servo axis is disabled and the drive decelerates to a minimum speed, the drive maintains torque until this time has elapsed. This time allows the motor's brake to be set. This attribute has a value range of 0...6.5535.
BrakeReleaseDelayTime	When the servo axis is enabled, the drive activates the torque to the motor but ignores the command values from the Logix controller until this time has elapsed. This time allows the motor's brake to release. This attribute has a value of 0...6.5535.
ResistiveBrakeContactDelay	The Resistive Brake Contact Delay attribute is used to control an optional external Resistive Brake Module (RBM). The RBM sits between the drive and the motor and uses an internal contactor to switch the motor between the drive and a resisted load.

Tag tab

Use this tab to modify the name and description of the axis. When you are online, all parameters on this tab transition to a read-only state, and cannot be modified. If you go online before you save your changes, all pending changes revert to their previously-saved state.



Name

Displays the name of the current tag. If desired, you can rename this tag.

Description

Displays the description of the current tag, if any is available. If desired, you can edit this description.

Tag Type

Indicates the type of the current tag. This type may be:

- Base
- Alias
- Consumed

Displays the data type associated with the current tag.

Data Type

Displays the axis data type of the current tag.

Scope

Displays the scope of the current tag. The scope is controller scope, or program scope, based on one of the programs in the controller.

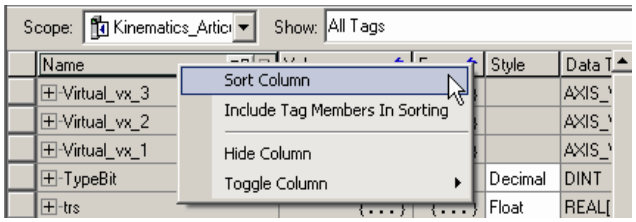
Style

Displays the default style in which to display the value of the tag. Note that style is only applicable to an atomic tag; a structure tag does not have a display style.

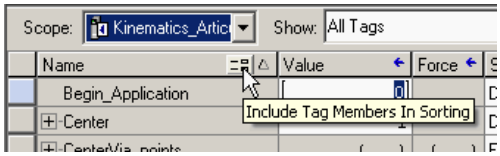
Monitoring axis tags

In Edit Tags or Monitor Tags, you can sort the tags alphabetically.

Right-click on the **Name** column and click **Sort Column**.

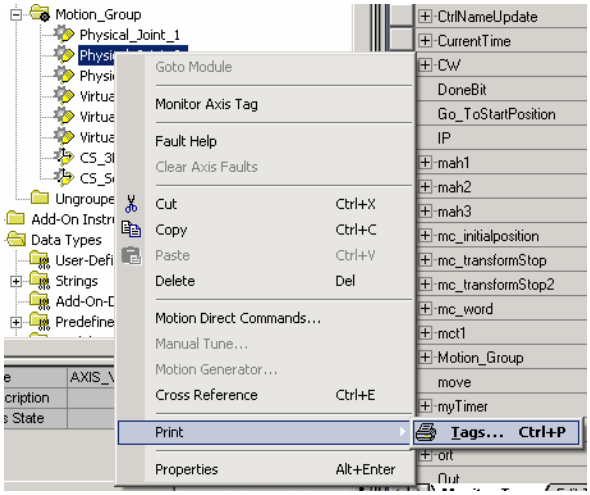


You also have the option to **Include Tag Members in Sorting**.

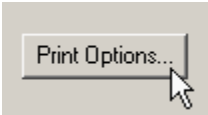


Create reports

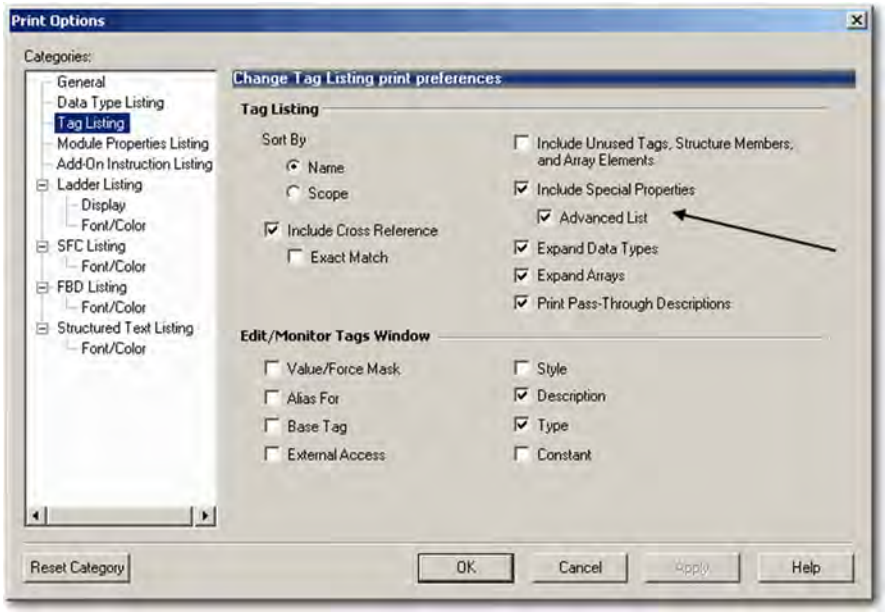
In Logix Designer application you are able to print a variety of reports. For example, right-click Controller Tags, MainTask, MainProgram, axis, Add-On Instructions, or Data Types.



On the Print dialog box, select the Adobe PDF and click **Print Options**.



When printing a Tag Listing, be sure to check the Include Special Properties and Advanced List to see the information.



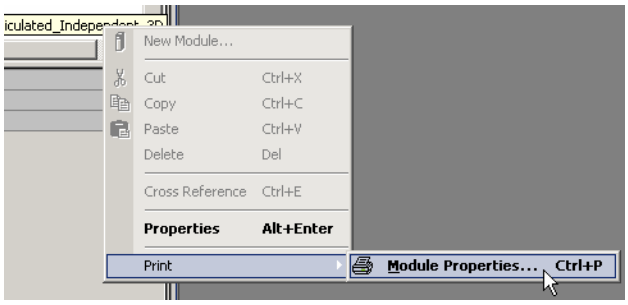
Axis Properties Example

Physical_Joint_3 - Axis Tag Properties **Page 2**
Kinematics_Articulated_Independent_3D (Controller) 1/27/2010 3:50:39 PM
C:\RSLogix 5000\Projects\Samples\ENU\v18\Rockwell Automation\Kinematics_Articulated_Independent_3D.ACD

Properties - Motion Axis tag: Physical_Joint_3

General			
Motion Group:	Motion_Group		
Motion Planner			
Output Cam Execution Targets:	0	Program Stop Action:	Fast Stop
Master Delay Compensation:	Yes		
Enable Master Position Filter:	No	Master Position Filter Bandwidth:	N/A
Units			
Position Units:	Inches	Average Velocity Timebase:	0.25 Seconds
Conversion			
Positioning Mode:	Linear	Conversion Constant:	25400.0 Feedback Counts/1.0 Inches
Position Unwind:	N/A		
Homing			
Mode:	Active	Position:	5.0 Inches
Sequence:	Immediate		
Dynamics			
Maximum Speed:	200000.0 Inches/s		
Maximum Acceleration:	2000000.0 Inches/s^2	Maximum Deceleration:	2000000.0 Inches/s^2
Maximum Acceleration Jerk:	60000000.0 Inches/s^3	Maximum Deceleration Jerk:	58431372.0 Inches/s^3

You can also right-click a controller, communication module, and any motion module to print the Module Properties you configured.



Module Properties Example

Local - Module Properties Listing **Page 1**
Kinematics_Articulated_Independent_3D: Local 1/27/2010 3:55:43 PM
C:\RSLogix 5000\Projects\Samples\ENU\v18\Rockwell Automation\Kinematics_Articulated_Independent_3D.ACD

1756 Backplane, 1756-A17 : Local Modules

Local: [0] 1756-L63 Kinematics_Articulated_Independent_3D

Type:	1756-L63 ControlLogix5563 Controller	Parent:	Local
Vendor:	Allen-Bradley	Vendor ID:	1
Slot:	0	Electronic Keying:	Exact Match
Revision:	18.1	Status:	Standby
Module Fault:	Offline	Inhibit Flag	Off

Motion axis attributes

Introduction for Motion Axis Attributes

The Access column shows how to access the attribute.

Attribute	Axis Type	Data Type	Access	Description
Acceleration Feedforward Gain			GSV	Use a Get System Value (GSV) instruction to get the value.
			SSV	Use a Set System Value (SSV) instruction to set or change the value. The attribute can only be modified when the axis is not enabled.
Accel Status			Tag	
Actual Acceleration			GSV	Use the tag for the axis to get the value.
			Tag	Use the tag for the axis or a GSV instruction to get the value. It's easier to use the tag.

Accessing an MSG instruction

For complete information on how to access data using an MSG instruction, see Logix5000 Controllers Messages Programming Manual, publication 1756-PM012.

If you need more information on Attribute and Class IDs, see the drive documentation.

Interpreting the Attribute Tables

The following table provides an explanation of the information, nomenclature, and abbreviations used in the attribute tables.

Column Heading	Description	
Attribute	Each attribute table begins with the attribute name. The tag, GSV/SSV, and MSG names for each of these attributes are the same as the attribute name listed, but with the spaces removed. For example, Absolute Feedback Enable would be AbsoluteFeedbackEnable.	
Axis Type	For each attribute, the related axis is listed.	
Data Type	The data type, such as DINT, UINT, SINT, REAL, and BOOL.	
Access	GSV	Can be read via the GSV instruction.
	SSV	Can be written via the SSV instruction.
	Tag	Can be created to allocate and reference data.
	MSG	Message is only used to access drive attributes for which there is not GSV/SSV access.
		To use a MSG instruction to access information from a drive, you will need the Attribute and Class IDs. See the drive documentation for ID information.
Description	The meaning of the attribute values, such as Position Units / Seconds. Tag access is supported, but value is valid only when Auto Tag Update of the Motion Group Object is enabled.	

Replicated Attributes

These are the controller attributes that are replicated in the motion module.

- AccelerationFeedForwardGain
- AxisType
- DriveFaultAction
- FeedbackFaultAction
- FeedbackNoiseFaultAction
- FrictionCompensation
- MaximumNegativeTravel
- MaximumPositiveTravel
- OutputLPFilterBandwidth
- OutputLimit
- OutputOffset

- PositionErrorFaultAction
- PositionErrorTolerance
- PositionIntegralGain
- PositionProportionalGain
- PositionUnwind
- SoftOvertravelFaultAction
- TorqueScaling
- VelocityFeedforwardGain
- VelocityIntegralGain
- VelocityProportionalGain
- VelocityScaling

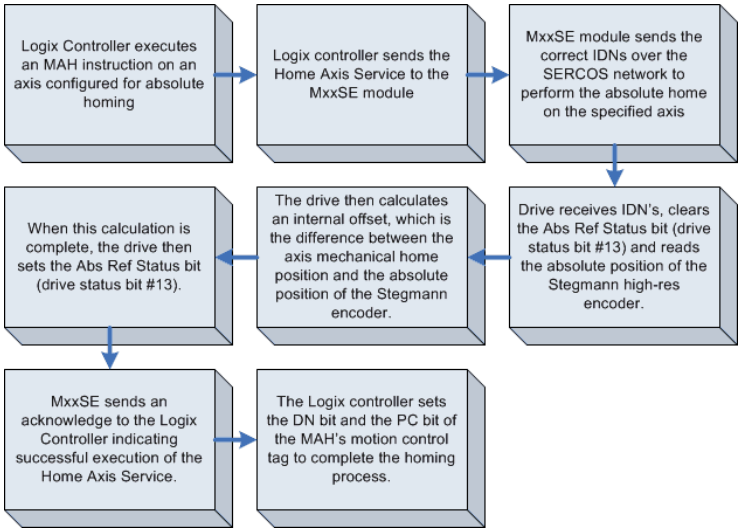
Axis attributes

This table describes each attribute of an axis.

Attribute	Axis Type	Data Type	Access	Description
Absolute Feedback Enable	AXIS_SERVO	SINT	GSV SSV	**Important:** Use this attribute only for an axis of a 1756-HYD02 or 1756-M02AS module. This attribute controls whether the servo module uses the absolute position capability of the feedback device. If Absolute Feedback Enable is set to True, the servo module adds the Absolute Feedback Offset to the current position of the feedback device to establish the absolute machine reference position. Since absolute feedback devices retain their position reference even through a power-cycle, the machine reference system can be restored at powerup. To establish a suitable value for the Absolute Feedback Offset attribute, the MAH instruction may be executed with the Home Mode configured for Absolute (the only option when Absolute Feedback Enable is True). When executed, the servo module computes the Absolute Feedback Offset as the difference between the configured value for Home Position and the current absolute feedback position of the axis. The computed Absolute Feedback Offset is immediately applied to the axis upon completion of the MAH instruction. Because the actual position of the axis is re-referenced during execution of the MAH instruction, the servo loop must not be active. If the servo loop is active the MAH instruction errors. If Absolute Feedback Enable is set to False, the servo module ignores the Absolute Feedback Offset and treats the feedback device as an incremental position transducer. In this case, a homing or redefine position operation is therefore needed to establish the absolute machine reference position. The Absolute Home Mode in this case is invalid. This attribute is configurable if the Transducer Type is set to SSI. For an LDT transducer, the Absolute Feedback Enable is forced to True. For an AQB transducer, the Absolute Feedback Enable is forced to False.

Attribute	Axis Type	Data Type	Access	Description
Absolute Feedback Offset	AXIS_SERVO	REAL	GSV SSV	Position Units **Important:** Use this attribute only for an axis of a 1756-HYD02 or 1756-M02AS module. Set the Absolute Feedback Enable attribute to True. This attribute is used to determine the relative distance between the absolute position of the feedback device and the absolute position of the machine. At powerup, this attribute is sent to the servo module and added to the current position of the feedback device to restore the absolute machine position reference. If the axis is configured for Linear operation, absolute position may be recovered after power cycle as long as the feedback device does not exceed its range limit. If the feedback device rolls over its count range, the absolute position of the axis is invalid. If the axis is configured for Rotary operation, the servo module is responsible for adjusting the Absolute Feedback Offset dynamically based on the configured Unwind value and the rollover of the absolute feedback device. If necessary, absolute position may be recovered after power cycle by periodically updating the controller's Absolute Feedback Offset value. This can be done by selecting the Absolute Feedback Offset enumeration for one of the Axis Info Select attributes.
Absolute Reference Status	AXIS_SERVO_DRIVE	BOOL	Tag	**If the bit is** / **Then** ON: An absolute homing procedure happened. The bit stays set until one of the following occurs. • The drive resets its configuration parameters to default values. • The axis does an active or passive home or redefine position. MRP also clears Absolute Reference Status OFF: The position of the axis is not referenced to the absolute machine reference system established by an absolute homing procedure.

Attribute	Axis Type	Data Type	Access	Description
Absolute Reference Status (continued)				The AbsoluteReferenceStatus bit provides an indication whether the system is absolutely referenced. When all conditions are configured correctly, the AbsoluteReferenceStatus remains set through a power cycle, and the absolute position remains intact. Bit = 1 when an absolute home occurred (see chapter 5 - guidelines for homing), along with the data listed below Bit = 0 when an absolute home did not occur, along with the data listed below **Conditions that cause the AbsoluteReferenceStatus bit to reset back to 0:** • Completing an MotionRedefinePosition (MRP) instruction (U3K & K6K) • Completing an MAH and specifying non-abs homing type • Replacing motor • Successful execution of a non-absolute MAH (U3K & K6K) • Offline, changing rotary to linear and vice versa and then downloading over the current configuration (U3K) • Changing Motor / Abs Feedback device • Power cycle to an axis with a single-turn feedback device configured as a linear axis (U3K) • Power cycle to a single-turn rotary axis with a Drive Resolution not equal to the Unwind (U3K & K6K) • E20 fault - Motor Encoder State Error. Motor encoder encountered an illegal state transition (U3K) • E32 fault - S/C Frequency Exceeded. Maximum frequency if hardware exceeded (U3K) • E73 Sercos Fault - Backplane Comm, Power Rail Backplane CAN communication failed. Typically a hardware failure or bent backplane pins (K6K) • E76 Drive Hard Fault - CAN bit, where DPI or Backplane CAN initialization failed. Typically a hardware failure or bent backplane pins (K6K) • Power cycle while auxiliary powered devices are producing excessive regenerative energy. Example: a fan or pump powered from the same supply powering aux power to the K6K • Reset drive to defaults using Drive Explorer (K6K) • Reset drive to defaults using Ultraware (U3K) • Transitioning from Ring phase 3 or Ring phase 4 if the encoder is not a multi-turn or single-turn absolute device (K6K & U3K) **A few conditions that do not cause the AbsoluteReferenceStatus bit to reset to 0:** • Feedback Loss, even through a power cycle. • Changing the unwind value • Battery replacement or low battery if control power remains active • Downloading Program Should stay intact after an upload or download as long the user uploads attribute to the offline image, once offline - otherwise, the offline image does not have the bit set. **Auxiliary Axis (feedback only) absolute capabilities - K6K (U3K)** Auxiliary axes have the same capabilities for maintaining an absolute reference as the main feedback, except the auxiliary axis channel cannot generate a marker from any sine / cosine device. This would include the SRS / SRM feedback devices **Types of Absolute devices allowed for "AxisType - Axis_Servo_Drive"** • Stegmann Hiperface SRM / SKM Encoder • Stegmann Hiperface SRS / SKS Encoder • Tamagawa TL5669 Encoder

Attribute	Axis Type	Data Type	Access	Description
Absolute Reference Status (continued)				**Sercos Absolute Homing Process Flowchart** 
Accel Limit Status	AXIS_SERVO_DRIVE	BOOL	Tag	Set when the magnitude of the commanded acceleration to the velocity servo loop input is greater than the configured Velocity Limit.
Accel Status	AXIS_CONSUMED AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE AXIS_VIRTUAL	BOOL	Tag	Set if the axis is currently being commanded to accelerate. Use the Accel Status bit and the Decel Status bit to see if the axis is accelerating or decelerating. If both bits are off, then the axis is moving at a steady speed or is at rest.
Acceleration Command	AXIS_SERVO AXIS_SERVO_DRIVE	REAL	GSV Tag	**Important:**To use this attribute, choose it as one of the attributes for Real Time Axis Information for the axis. Otherwise, you won't see the right value as the axis runs. See Axis Info Select 1. Acceleration Command in Position Units / Sec^2 Acceleration Command is the current acceleration reference to the output summing junction, in the configured axis Position Units per $Second^2$, for the specified axis. The Acceleration Command value, hence, represents the output of the inner velocity control loop. Acceleration Command is not to be confused with Command Velocity, which represents the rate of change of Command Position input to the position servo loop.
Acceleration Data Scaling	AXIS_SERVO_DRIVE	INT	GSV	This attribute is derived from the Drive Units attribute. See IDN 160 in IEC 1491.
Acceleration Data Scaling Exp	AXIS_SERVO_DRIVE	INT	GSV	This attribute is derived from the Drive Units attribute. See IDN 162 in IEC 1491.
Acceleration Data Scaling Factor	AXIS_SERVO_DRIVE	DINT	GSV	This attribute is derived from the Drive Units attribute. See IDN 161 in IEC 1491.
Acceleration Feedback	AXIS_SERVO AXIS_SERVO_DRIVE	REAL	GSV Tag	**Important:** To use this attribute, choose it as one of the attributes for Real Time Axis Information for the axis. Otherwise, you won't see the right value as the axis runs. See Axis Info Select 1. Acceleration Feedback in Position Units / Sec^2 Acceleration Feedback is the actual velocity of the axis as estimated by the servo module, in the configured axis Position Units per $Second^2$. The Estimated Acceleration is calculated by taking the difference in the Estimated Velocity over the servo update interval. Acceleration Feedback is a signed value—the sign (+ or -) depends on which direction the axis is currently moving.

Attribute	Axis Type	Data Type	Access	Description
Acceleration Feedforward Gain	AXIS_SERVO AXIS_SERVO_DRIVE	REAL	GSV SSV	% This controller attribute is also replicated in the motion module. **AXIS_SERVO** When you connect to a torque servo drive, use the Acceleration Feedforward Gain to give the Torque Command output necessary to generate the commanded acceleration. It does this by scaling the current Command Acceleration by the Acceleration Feedforward Gain and adding it as an offset to the Servo Output generated by the servo loop. With this done, the servo loops do not need to generate much of a contribution to the Servo Output, hence the Position and/or Velocity Error values are significantly reduced. Hence, when used in conjunction with the Velocity Feedforward Gain, the Acceleration Feedforward Gain lets the following error of the servo system during the acceleration and deceleration phases of motion be reduced to nearly zero. This is important in applications such as electronic gearing and synchronization where the actual axis position must not significantly lag behind the commanded position at any time. When you connect to a velocity servo drive, use Acceleration Feedforward to add a term to the Velocity Command that is proportional to the commanded acceleration. This can be effective in cases where the external drive shows a steady-state velocity error during acceleration and deceleration. The best value for Acceleration Feedforward depends on the drive configuration. Excessive Acceleration Feedforward values tend to produce axis overshoot. For torque servo drive applications, the best value for Acceleration Feedforward is theoretically 100%. However, the value may need to be increased slightly to accommodate servo loops with non-infinite loop gain and other application considerations. For velocity servo drive applications, the best value for Acceleration Feedforward is highly dependent on the drive's speed scaling and servo loop configuration. A value of 100%, in this case, means only that 100% of the commanded acceleration value is applied to the velocity command summing junction and may not be even close to the optimal value. To find the best Acceleration Feedforward Gain, run a simple project that jogs the axis in the positive direction and monitors the Position Error of the axis during the jog. Usually Acceleration Feedforward is used in tandem with Velocity Feedforward to achieve near zero following error during the entire motion profile. To fine tune the Acceleration Feedforward Gain, the Velocity Feedforward Gain must first be optimized using the procedure described above. While capturing the peak Position Error during the acceleration phase of the jog profile, increase the Acceleration Feedforward Gain until the peak Position Error is as small as possible, but still positive. If the peak Position Error during the acceleration ramp is negative, the actual position of the axis is *ahead* of the command position during the acceleration ramp. If this occurs, decrease the Acceleration Feedforward Gain such that the Position Error is again positive. To be thorough, the same procedure should be done for the deceleration ramp to verify that the peak Position Error during deceleration is acceptable. Note that reasonable maximum velocity, acceleration, and deceleration values must be entered to jog the axis.

Attribute	Axis Type	Data Type	Access	Description
Acceleration Feedforward Gain (continued)				**AXIS_SERVO_DRIVE** The Acceleration Feedforward Gain attribute is used to provide the Torque Command output necessary to generate the commanded acceleration. It does this by scaling the current Command Acceleration by the Acceleration Feedforward Gain and adding it as an offset to the Servo Output generated by the servo loop. With this done, the servo loops do not need to generate much control effort, hence the Position and/or Velocity Error values are significantly reduced. When used in conjunction with the Velocity Feedforward Gain, the Acceleration Feedforward Gain allows the following error of the servo system during the acceleration and deceleration phases of motion to be reduced to nearly zero. This is important in applications such as electronic gearing and synchronization applications where it is necessary that the actual axis position not significantly lag behind the commanded position at any time. The optimal value for Acceleration Feedforward is 100% theoretically. In reality, however, the value may need to be tweaked to accommodate torque loops with non-infinite loop gain and other application considerations. One thing that may force a smaller Acceleration Feedforward value is that increasing amounts of feedforward tends to exacerbate axis overshoot. When necessary, the Acceleration Feedforward Gain may be 'tweaked' from the 100% value by running a simple user program that jogs the axis in the positive direction and monitors the Position Error of the axis during the jog. Usually Acceleration Feedforward is used in tandem with Velocity Feedforward to achieve near zero following error during the entire motion profile. To fine-tune the Acceleration Feedforward Gain, the Velocity Feedforward Gain must first be optimized using the procedure described above. While capturing the peak Position Error during the acceleration phase of the jog profile, increase the Acceleration Feedforward Gain until the peak Position Error is as small as possible, but still positive. If the peak Position Error during the acceleration ramp is negative, the actual position of the axis is *ahead* of the command position during the acceleration ramp. If this occurs, decrease the Acceleration Feedforward Gain such that the Position Error is again positive. To be thorough the same procedure should be done for the deceleration ramp to verify that the peak Position Error during deceleration is acceptable. Note that reasonable maximum velocity, acceleration, and deceleration values must be entered to jog the axis.
Acceleration Limit Bipolar	AXIS_SERVO_DRIVE	REAL	GSV SSV	Position Units / Sec^2 This attribute maps directly to a SERCOS IDN. See the SERCOS Interface standard for a description. This attribute is automatically set. You usually do not need to change it.
Acceleration Limit Negative	AXIS_SERVO_DRIVE	REAL	GSV SSV	Position Units / Sec^2 This attribute maps directly to a SERCOS IDN. See the SERCOS Interface standard for a description. This attribute is automatically set. You usually do not need to change it.

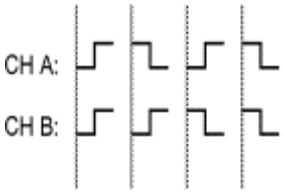
Attribute	Axis Type	Data Type	Access	Description
Acceleration Limit Positive	AXIS_SERVO_DRIVE	REAL	GSV SSV	Position Units / Sec^2 This attribute maps directly to a SERCOS IDN. See the SERCOS Interface standard for a description. This attribute is automatically set. You usually do not need to change it.
Actual Acceleration	AXIS_CONSUMED AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE AXIS_VIRTUAL	REAL	GSV Tag	**Important:** To use this attribute, make sure Auto Tag Update is Enabled for the motion group (default setting). Otherwise, you won't see the right value as the axis runs. Actual Acceleration in Position Units / Sec^2 Actual Acceleration is the current instantaneously measured acceleration of an axis, in the configured axis Position Units per second per second. It is calculated as the current increment to the actual velocity per base update interval. Actual Acceleration is a signed value — the sign (+ or -) depends on which direction the axis is currently accelerating. Actual Acceleration is a signed floating-point value. Its resolution does not depend on the Averaged Velocity Timebase, but rather on the conversion constant of the axis and the fact that the internal resolution limit on actual velocity is 1 feedback counts per base update period per base update period.
Actual Position	AXIS_CONSUMED AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE AXIS_VIRTUAL	REAL	GSV Tag	**Important:** To use this attribute, make sure Auto Tag Update is Enabled for the motion group (default setting). Otherwise, you won't see the right value as the axis runs. Actual Position in Position Units Actual Position is the current absolute position of an axis, in the configured Position Units of that axis, as read from the feedback transducer. Note, however, that this value is based on data reported to the controller as part of an ongoing synchronous data transfer process, which results in a delay of one base update period. Thus, the Actual Position value that is obtained is the actual position of the axis one base update period ago.
Actual Velocity	AXIS_CONSUMED AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE AXIS_VIRTUAL	REAL	GSV Tag	**Important:** To use this attribute, make sure Auto Tag Update is Enabled for the motion group (default setting). Otherwise, you won't see the right value as the axis runs. Actual Velocity in Position Units / Sec Actual Velocity is the current instantaneously measured speed of an axis, in the configured axis Position Units per second. It is calculated as the current increment to the actual position per base update interval. Actual Velocity is a signed value—the sign (+ or -) depends on which direction the axis is currently moving. Actual Velocity is a signed floating-point value. Its resolution does not depend on the Averaged Velocity Timebase, but rather on the conversion constant of the axis and the fact that the internal resolution limit on actual velocity is 1 feedback counts per base update.
Analog Input1 Analog Input 2	AXIS_SERVO_DRIVE	REAL	GSV SSV	This attribute applies only to an axis associated with a Kinetix 7000 drive. The integer range is +/-16384, representing the analog value of an analog device connected to the Kinetix 7000 drive analog input. These inputs are useful for web/converting applications with load cell (measuring web force on a roller) or dancer (measuring web force/position directly) that can be directly connected to the drive controlling the web.

Attribute	Axis Type	Data Type	Access	Description
Attribute Error Code	AXIS_SERVO AXIS_SERVO_DRIVE	INT	GSV* Tag	CIP Error code returned by erred set attribute list service to the module. When an Axis Configuration Fault occurs, one or more axis parameters associated with a motion module or device did not successfully update to match the value of the corresponding parameter of the local controller. The fact that the configuration of the axis no longer matches the configuration of the local controller is a serious fault and results in the shutdown of the faulted axis. The Attribute Error Code is reset to zero by reconfiguration of the motion module. Axis Configuration Fault information is passed from the motion module or device to the controller via a 16-bit CIP status word contained in the Set Attribute List service response received by the controller. A Set Attribute List service to the motion module can be initiated by a software Set Attribute List service to the controller, or by an SSV instruction within the controller's program, referencing a servo attribute. Various routines that process responses to motion services are responsible for updating these attributes. The Set and Get service responses provide a status response with each attribute that was processed. That status value is defined by CIP as follows: UINT16, Values 0-255 (0x00-0xFF) are reserved to mirror common service status codes. Values 256 – 65535 are available for object/class attribute errors.
Attribute Error ID	AXIS_SERVO AXIS_SERVO_DRIVE	INT	GSV* Tag	Attribute ID associated with non-zero Attribute Error Code. The Attribute Error ID is used to retain the ID of the servo attribute that returned a non-zero attribute error code resulting in an Axis Configuration Fault. The Attribute Error ID defaults to zero and, after a fault occurs, may be reset to zero by reconfiguration of the motion module. To quickly see the Attribute Error in Logix Designer application: Select the axis in the Controller Organizer. Examine the bottom of the Controller Organizer for the Attribute Error.

<table>
<tr><th>Attribute</th><th>Axis Type</th><th>Data Type</th><th>Access</th><th>Description</th></tr>
<tr><td>Aux Feedback Configuration</td><td>AXIS_SERVO_DRIVE</td><td>INT</td><td>GSV</td><td>The controller and drive use this for scaling the feedback device counts. These attributes are derived from the corresponding Motor and Auxiliary Feedback Unit attributes.
Bit
0 = Feedback type
0 — rotary (default)
1 — linear
1 = (reserved)
2 = Linear feedback unit
0 — metric
1 — English
3 = Feedback Polarity (Aux Only)
0 — not inverted
1 — inverted

<table><tr><th colspan="3">If the bits are</th><th>Then Feedback Resolution is scaled to</th></tr><tr><th>2</th><th>1</th><th>0</th><th></th></tr><tr><td>0</td><td></td><td>0</td><td>Feedback Cycles per Feedback Rev</td></tr><tr><td>1</td><td></td><td>0</td><td>Feedback Cycles per Feedback Rev</td></tr><tr><td>0</td><td></td><td>1</td><td>Feedback Cycles per mm</td></tr><tr><td>1</td><td></td><td>1</td><td>Feedback Cycles per inch</td></tr></table>
Feedback Polarity
The Feedback Polarity bit attribute can be used to change the sense of direction of the feedback device. This bit is only valid for auxiliary feedback devices. When performing motor/feedback hookup diagnostics on an auxiliary feedback device using the MRHD and MAHD instructions, the Feedback Polarity bit is configured for the auxiliary feedback device to insure negative feedback into the servo loop. Motor feedback devices must be wired properly for negative feedback since the Feedback Polarity bit is forced to 0, or non-inverted.</td></tr>
<tr><td>Aux Feedback Fault</td><td>AXIS_SERVO
AXIS_SERVO_DRIVE</td><td>BOOL</td><td>Tag</td><td>Set for an auxiliary feedback source when one of these happens:
The differential electrical signals for one or more of the feedback channels (for example, A+ and A-, B+ and B-, or Z+ and Z-) are at the same level (high or low). Under normal operation, the differential signals are at opposite levels. The most common cause of this situation is a broken wire between the feedback transducer and the servo module or drive;
Loss of feedback ‘power’ or feedback ‘common’ electrical connection between the servo module or drive and the feedback device.
The controller latches this fault. Use a Motion Axis Fault Reset (MAFR) or Motion Axis Shutdown Reset (MASR) instruction to clear the fault.</td></tr>
<tr><td>Aux Feedback Interpolation Factor</td><td>AXIS_SERVO_DRIVE</td><td>DINT</td><td>GSV</td><td>Feedback Counts per Cycle
The Feedback Interpolation attributes establish how many Feedback Counts there are in one Feedback Cycle. The Feedback Interpolation Factor depends on the feedback device and the drive feedback circuitry. Quadrature encoder feedback devices and the associated drive feedback interface typically support 4x interpolation, so the Interpolation Factor for these devices would be set to 4 Feedback Counts per Cycle (Cycles are sometimes called Lines). High Resolution Sin/Cosine feedback device types can have interpolation factors as high as 2048 Counts per Cycle. The product of the Feedback Resolution and the corresponding Feedback Interpolation Factor is the overall resolution of the feedback channel in Feedback Counts per Feedback Unit. In our example, a Quadrature encoder with a 2000 line/rev resolution and 4x interpolation factor would have an overall resolution of 8000 counts/rev.</td></tr>
</table>

Attribute	Axis Type	Data Type	Access	Description

Aux Feedback Noise Fault	AXIS_SERVO_DRIVE	BOOL	Tag	Set when there is noise on the feedback device's signal lines.
				• For example, simultaneous transitions of the feedback A and B channels of an A Quad B is referred to generally as feedback noise.
				• Feedback noise (shown below) is most often caused by loss of quadrature in the feedback device itself or radiated common-mode noise signals being picked up by the feedback device wiring. You can see these on an oscilloscope.
				 
				• To troubleshoot the loss of channel quadrature, look for:
				  • physical misalignment of the feedback transducer components
				  • excessive capacitance (or other delays) on the encoder signals
				• Proper grounding and shielding usually cures radiated noise problems.
				The controller latches this fault. Use a Motion Axis Fault Reset (MAFR) or Motion Axis Shutdown Reset (MASR) instruction to clear the fault.

Aux Feedback Ratio	AXIS_SERVO_DRIVE	FLOAT	GSV	Aux Feedback Units per Motor Feedback Unit The Aux Feedback Ratio attribute represents the quantitative relationship between auxiliary feedback device and the motor. For a rotary auxiliary feedback device, this attribute's value should be the turn ratio between the auxiliary feedback device and the motor shaft. For linear auxiliary feedback devices, this attribute value would typically represent the feed constant between the motor shaft and the linear actuator. The Aux Feedback Ratio attribute is used in calculating range limits and default value calculations during configuration based on the selected motor's specifications. The value is also used by the drive when running the dual feedback servo loop configuration.
Aux Feedback Resolution	AXIS_SERVO_DRIVE	DINT	GSV	Cycles per Aux Feedback Unit The Motor and Aux Feedback Resolution attributes are used to provide the A-B drive with the resolution of the associated feedback device in cycles per feedback unit. These parameters provide the SERCOS drive with critical information needed to compute scaling factors used to convert Drive Counts to Feedback counts.

<table>
<tr><th>Attribute</th><th>Axis Type</th><th>Data Type</th><th>Access</th><th>Description</th></tr>
<tr><td>Aux Feedback Type</td><td>AXIS_SERVO_DRIVE</td><td>INT</td><td>GSV</td><td>The Motor and Aux Feedback Type attributes are used to identify the motor mounted or auxiliary feedback device connected to the drive.
<table>
<tr><th>Feedback Type</th><th>Code</th><th>Rotary Only</th><th>Linear Only</th><th>Rotary or Linear</th></tr>
<tr><td><None></td><td>0x0000</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>SRS</td><td>0x0001</td><td>X</td><td></td><td></td></tr>
<tr><td>SRM</td><td>0x0002</td><td>X</td><td></td><td></td></tr>
<tr><td>SCS</td><td>0x0003</td><td>X</td><td></td><td></td></tr>
<tr><td>SCM</td><td>0x0004</td><td>X</td><td></td><td></td></tr>
<tr><td>SNS</td><td>0x0005</td><td>X</td><td></td><td></td></tr>
<tr><td>MHG</td><td>0x0006</td><td>X</td><td></td><td></td></tr>
<tr><td>Resolver</td><td>0x0007</td><td>X</td><td></td><td></td></tr>
<tr><td>Analog Reference</td><td>0x0008</td><td>X</td><td></td><td></td></tr>
<tr><td>Sin/Cos</td><td>0x0009</td><td></td><td></td><td>X</td></tr>
<tr><td>TTL</td><td>0x000A</td><td></td><td></td><td>X</td></tr>
<tr><td>UVW</td><td>0x000B</td><td></td><td></td><td>X</td></tr>
<tr><td>Unknown Stegmann</td><td>0x000C</td><td></td><td></td><td>X</td></tr>
<tr><td>Endat</td><td>0x000D</td><td></td><td></td><td>X</td></tr>
<tr><td>RCM21S-4</td><td>0x000E</td><td>X</td><td></td><td></td></tr>
<tr><td>RCM21S-6</td><td>0x000F</td><td>X</td><td></td><td></td></tr>
<tr><td>RCM21S-8</td><td>0x0010</td><td>X</td><td></td><td></td></tr>
<tr><td>LINCODER</td><td>0x0011</td><td></td><td>X</td><td></td></tr>
<tr><td>Sin/Cos with Hall</td><td>0x0012</td><td></td><td></td><td>X</td></tr>
<tr><td>TTL with Hall</td><td>0x0013</td><td></td><td></td><td>X</td></tr>
</table></td></tr>
<tr><td>Aux Feedback Units</td><td>AXIS_SERVO_DRIVE</td><td>INT</td><td>GSV</td><td>The Motor Feedback Units attribute establishes the unit of measure that is applied to the Motor Feedback Resolution attribute value. The Aux Feedback Units attribute establishes the unit of measure that is applied to the Aux Feedback Resolution attribute value. Units appearing in the enumerated list cover linear or rotary, English, or metric feedback devices.
0 = revs
1 = inches
2 = mm</td></tr>
<tr><td>Aux Position Feedback</td><td>AXIS_SERVO
AXIS_SERVO_DRIVE</td><td>REAL</td><td>GSV
Tag</td><td>Important: To use this attribute, choose it as one of the attributes for Real Time Axis Information for the axis. Otherwise, you won't see the right value as the axis runs. See Axis Info Select 1.
Auxiliary Position Feedback in Position Units
Aux Position Feedback is the current value of the position feedback coming from the auxiliary feedback input.</td></tr>
</table>

Attribute	Axis Type	Data Type	Access	Description
Average Velocity	AXIS_CONSUMED AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE AXIS_VIRTUAL	REAL	GSV Tag	**Important:** To use this attribute, make sure Auto Tag Update is Enabled for the motion group (default setting). Otherwise, you won't see the right value as the axis runs. Average Velocity in Position Units / Seconds Average Velocity is the current speed of an axis in the configured Position Units per second of the axis. Unlike the Actual Velocity attribute value, it is calculated by averaging the actual velocity of the axis over the configured Average Velocity Timebase for that axis. Average velocity is a signed value. The sign does not necessarily show the direction that the axis is currently moving. It shows the direction the average move is going. The axis may be currently moving in the opposite direction. The resolution of the Average Velocity variable is determined by the current value of the Averaged Velocity Timebase parameter and the configured Conversion Constant (feedback counts per Position Unit) for the axis. • The greater the Average Velocity Timebase value, the better the speed resolution, but the slower the response to changes in speed. • The minimum Average Velocity Timebase value is the Base Update period of the motion group. The Average Velocity resolution in Position Units per second may be calculated using the equation below. $\text{Average Velocity Timebase [Seconds]} \times K \left[\frac{\text{Feedback Counts}}{\text{Position Unit}}\right]$ For example, on an axis with position units of inches and a conversion constant (K) of 20000, an averaged velocity time-base of 0.25 seconds results in an average velocity resolution of: $\frac{1}{0.25 \times 20000} = 0.0002 \frac{\text{Inches}}{\text{Second}} = 0.012 \frac{\text{Inches}}{\text{Minute}}$
Average Velocity Timebase	AXIS_CONSUMED AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE AXIS_VIRTUAL	REAL	GSV SSV	Seconds The Average Velocity Timebase attribute is used to specify the desired time in seconds to be used for calculating the Average Velocity of the axis. When the Average Velocity Value is requested, the value is computed by taking the total distance traveled by the axis in the amount of time given by the Average Velocity Timebase and dividing this value by the timebase. The Average Velocity Timebase value should be large enough to filter out the small changes in velocity that would otherwise result in a 'noisy' velocity value, but small enough to track significant changes in axis velocity. Typically, a value between 0.25 and 0.5 seconds works well for most applications
Axis Configuration State	AXIS_CONSUMED AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE AXIS_VIRTUAL	SINT	GSV	State of the axis configuration state machine The Axis Configuration State attribute is used for debugging to indicate where in the axis configuration state-machine this axis presently is. Even consumed and virtual axes use this attribute. If the attribute is: 128 — the axis is configured and ready for use. Not 128 — the axis is not configured.

Attribute	Axis Type	Data Type	Access	Description
Axis Control Bits	AXIS_SERVO AXIS_SERVO_DRIVE	DINT	GSV	Bits 0 = Abort Process Request 1 = Shutdown Request 2 = Zero DAC Request 3 = Abort Home Request 4 = Abort Event Request 5-14 = Reserved 15 = Change Cmd Reference **Abort Process** If this bit is set, any active tuning or test process on the axis is aborted. **Shutdown Request** If this bit is set, the axis is forced into the shutdown state. For an AXIS_SERVO data type, the OK contact opens and the DAC output goes to 0. **Zero DAC Request — Only for AXIS_SERVO Data Type** If this bit is set, the servo module forces the DAC output for the axis to zero volts. This bit only applies if the axis is in the Direct Drive State with the drive enabled but no servo action. **Abort Home Request** If this bit is set, any active homing procedures are canceled. **Abort Event Request** If this bit is set, any active registration or watch event procedures are canceled. **Change Cmd Reference** If this bit is set, the controller switches to a new position coordinate system for command position. The servo module or drive uses this bit when processing new command position data from the controller to account for the offset implied by the shift in the reference point. The bit is cleared when the axis acknowledges completion of the reference position change by clearing its Change Position Reference bit.

<table>
<tr><th>Attribute</th><th>Axis Type</th><th>Data Type</th><th>Access</th><th>Description</th></tr>
<tr><td>Axis Data Type</td><td>AXIS_CONSUMED
AXIS_GENERIC
AXIS_SERVO
AXIS_SERVO_DRIVE
AXIS_VIRTUAL</td><td>SINT</td><td>MSG</td><td>Associated motion axis tag data type:
0 = Feedback
1 = Consumed
2 = Virtual
3 = Generic
4 = Servo
5 = Servo Drive
6 = Generic Drive
The Axis Data Type attribute and is used to determine which data template, memory format, and set of attributes are created and applicable for this axis instance. This attribute can only be set as part of an axis create service.
Feedback
A feedback-only axis associated with feedback-only modules like PLS II and CFE, supporting quadrature encoder, resolver, HiperFace, and so on.
Consumed
A consumed axis that consumes axis motion data produced by a motion axis on another controller.
Virtual
A virtual axis having full motion planner operation but not associated with any physical device.
Generic
An axis with full motion planner functionality but no integrated configuration support; associated with devices such as DriveLogix, 1756-DM.
Servo
An axis with full motion planner functionality and integrated configuration support; associated with modules closing a servo loop and sending an analog command to an external drive; that is, 1756-M02AE, 1756-HYD02, and 1756-M02AS modules.
Servo Drive
An axis with full motion planner functionality and integrated configuration support; associated with digital drive interface modules sending a digital command to the external drive; that is, 1756-M03SE, 1756-M08SE, and 17556-M16SE (SERCOS interface).
Generic Drive
An axis of a SERCOS interface drive that is Extended Pack Profile compliant and on the ring of a 1756-M08SEG module.</td></tr>
<tr><td>Axis Event</td><td>AXIS_CONSUMED
AXIS_GENERIC
AXIS_SERVO
AXIS_SERVO_DRIVE
AXIS_VIRTUAL</td><td>DINT</td><td>Tag</td><td>Allows access to all event status bits in one 32-bit word. This tag is the same as the Axis Event Bits attribute.
<table><tr><th>Event Status</th><th>Bit</th></tr><tr><td>Watch Event Armed Status</td><td>0</td></tr><tr><td>Watch Event Status</td><td>1</td></tr><tr><td>Reg Event 1 Armed Status</td><td>2</td></tr><tr><td>Reg Event 1 Status</td><td>3</td></tr><tr><td>Reg Event 2 Armed Status</td><td>4</td></tr><tr><td>Reg Event 2 Status</td><td>5</td></tr><tr><td>Home Event Armed Status</td><td>6</td></tr><tr><td>Home Event Status</td><td>7</td></tr></table></td></tr>
</table>

<table>
<tr><th>Attribute</th><th>Axis Type</th><th>Data Type</th><th>Access</th><th>Description</th></tr>
<tr><td>Axis Event Bits</td><td>AXIS_CONSUMED
AXIS_GENERIC
AXIS_SERVO
AXIS_SERVO_DRIVE
AXIS_VIRTUAL</td><td>DINT</td><td>GSV</td><td>Allows access to all event status bits in one 32-bit word. This attribute is the same as the Axis Event tag.
<table><tr><th>Event Status</th><th>Bit</th></tr><tr><td>Watch Event Armed Status</td><td>0</td></tr><tr><td>Watch Event Status</td><td>1</td></tr><tr><td>Reg Event 1 Armed Status</td><td>2</td></tr><tr><td>Reg Event 1 Status</td><td>3</td></tr><tr><td>Reg Event 2 Armed Status</td><td>4</td></tr><tr><td>Reg Event 2 Status</td><td>5</td></tr><tr><td>Home Event Armed Status</td><td>6</td></tr><tr><td>Home Event Status</td><td>7</td></tr></table></td></tr>
<tr><td>Axis Fault</td><td>AXIS_CONSUMED
AXIS_GENERIC
AXIS_SERVO
AXIS_SERVO_DRIVE
AXIS_VIRTUAL</td><td>DINT</td><td>Tag</td><td>The axis faults for your axis:
<table><tr><th>Type of Fault</th><th>Bit</th></tr><tr><td>Physical Axis Fault</td><td>0</td></tr><tr><td>Module Fault</td><td>1</td></tr><tr><td>Config Fault</td><td>2</td></tr></table>This attribute is the same as the Axis Fault Bits attribute.</td></tr>
<tr><td>Axis Fault Bits</td><td>AXIS_CONSUMED
AXIS_GENERIC
AXIS_SERVO
AXIS_SERVO_DRIVE
AXIS_VIRTUAL</td><td>DINT</td><td>GSV*</td><td>The axis faults for your axis:
<table><tr><th>Type of Fault</th><th>Bit</th></tr><tr><td>Physical Axis Fault</td><td>0</td></tr><tr><td>Module Fault</td><td>1</td></tr><tr><td>Config Fault</td><td>2</td></tr></table>This attribute is the same as the Axis Fault tag.</td></tr>
</table>

<table>
<tr><th>Attribute</th><th>Axis Type</th><th>Data Type</th><th>Access</th><th>Description</th></tr>
<tr><td>Axis Info Select 1
Axis Info Select 2</td><td>AXIS_SERVO
AXIS_SERVO_DRIVE</td><td>DINT</td><td>GSV
SSV</td><td>There is a group of attributes that do not get updated by default.
• To use one of them, you must choose it for Real Time Axis Information for the axis. Otherwise, its value won't change and you won't see the right value as the axis runs.
• You can choose up to 2 of these attributes.
To use a GSV instruction to choose an attribute for Real Time Axis Information, set the Axis Info Select 1 or Axis Info Select 2 attribute.
<table>
<tr><th>AXIS_SERVO</th><th>AXIS_SERVO_DRIVE</th><th>Value</th></tr>
<tr><td>None (default)</td><td>None (default)</td><td>0</td></tr>
<tr><td>Position Command</td><td>Position Command</td><td>1</td></tr>
<tr><td>Position Feedback</td><td>Position Feedback</td><td>2</td></tr>
<tr><td>Aux Position Feedback</td><td>Aux Position Feedback</td><td>3</td></tr>
<tr><td>Position Error</td><td>Position Error</td><td>4</td></tr>
<tr><td>Position Integrator Error</td><td>Position Integrator Error</td><td>5</td></tr>
<tr><td>Velocity Command</td><td>Velocity Command</td><td>6</td></tr>
<tr><td>Velocity Feedback</td><td>Velocity Feedback</td><td>7</td></tr>
<tr><td>Velocity Error</td><td>Velocity Error</td><td>8</td></tr>
<tr><td>Velocity Integrator Error</td><td>Velocity Integrator Error</td><td>9</td></tr>
<tr><td>Acceleration Command</td><td>Acceleration Command</td><td>10</td></tr>
<tr><td>Acceleration Feedback</td><td>Acceleration Feedback</td><td>11</td></tr>
<tr><td>Servo Output Level</td><td></td><td>12</td></tr>
<tr><td>Marker Distance</td><td>Marker Distance</td><td>13</td></tr>
<tr><td></td><td>Torque Command</td><td>14</td></tr>
<tr><td></td><td>Torque Feedback</td><td>15</td></tr>
<tr><td></td><td>Positive Dynamic Torque Limit</td><td>16</td></tr>
<tr><td></td><td>Negative Dynamic Torque Limit</td><td>17</td></tr>
<tr><td></td><td>Motor Capacity</td><td>18</td></tr>
<tr><td></td><td>Drive Capacity</td><td>19</td></tr>
<tr><td></td><td>Power Capacity</td><td>20</td></tr>
<tr><td></td><td>Bus Regulator Capacity</td><td>21</td></tr>
<tr><td></td><td>Motor Electrical Angle</td><td>22</td></tr>
<tr><td></td><td>Torque Limit Source</td><td>23</td></tr>
<tr><td></td><td>DC Bus Voltage</td><td>24</td></tr>
<tr><td>Absolute Offset</td><td></td><td>25</td></tr>
<tr><td></td><td></td><td>26</td></tr>
<tr><td></td><td></td><td>27</td></tr>
<tr><td></td><td>Guard Status</td><td>28</td></tr>
<tr><td></td><td>Guard Faults</td><td>29</td></tr>
</table></td></tr>
</table>

Attribute	Axis Type	Data Type	Access	Description
Axis Instance	AXIS_CONSUMED AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE AXIS_VIRTUAL	INT	GSV	Instance Number assigned to Axis The Axis Instance attribute is used to return the instance number of an axis. Major fault records generated for an axis major fault contains only the instance of the offending axis. This attribute would then typically be used by a user to determine if this was the offending axis; that is, if the instance number matches.
Axis Response Bits	AXIS_SERVO AXIS_SERVO_DRIVE	DINT	GSV	Bits 0 = Abort Process Acknowledge 1 = Shutdown Acknowledge 2 = Zero DAC Acknowledge 3 = Abort Home Acknowledge 4 = Abort Event Acknowledge 5…14 = Reserved 15 = Change Pos Reference **Abort Process Acknowledge** If this bit is set, the tuning or test process was aborted. **Shutdown Acknowledge** If this bit is set, the axis was forced into the shutdown state. **Zero DAC Acknowledge — Only for AXIS_SERVO Data Type** If this bit is set, the DAC output for the axis was set to zero volts. **Abort Home Acknowledge** If this bit is set, the active home procedure was aborted. **Abort Event Acknowledge** If this bit is set, the active registration or watch position event procedure was aborted. **Change Pos Reference** If this bit is set, the Servo loop switched to a new position coordinate system. The controller uses this bit when processing new position data from the servo module or drive to account for the offset implied by the shift in the reference point. The bit is cleared when the controller acknowledges completion of the reference position change by clearing its Change Cmd Reference bit.
Axis State	AXIS_CONSUMED AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE AXIS_VIRTUAL	SINT	GSV	Operating state of the axis. 0 = Axis Ready 1 = Direct Drive Control 2 = Servo Control 3 = Axis Faulted 4 = Axis Shutdown 5 = Axis Inhibited 6 = Axis Ungrouped 7 = No Module 8 = Configuring

<table>
<tr><th>Attribute</th><th>Axis Type</th><th>Data Type</th><th>Access</th><th>Description</th></tr>
<tr><td>Axis Status</td><td>AXIS_CONSUMED
AXIS_GENERIC
AXIS_SERVO
AXIS_SERVO_DRIVE
AXIS_VIRTUAL</td><td>DINT</td><td>Tag</td><td>Allows access to all axis status bits in one 32-bit word. This tag is the same as the Axis Status Bits attribute.<table><tr><th>Axis Status</th><th>Bit</th></tr><tr><td>Servo Action Status</td><td>0</td></tr><tr><td>Drive Enable Status</td><td>1</td></tr><tr><td>Shutdown Status</td><td>2</td></tr><tr><td>Config Update In Process</td><td>3</td></tr><tr><td>Inhibit Status</td><td>4</td></tr></table></td></tr>
<tr><td>Axis Status Bits</td><td>AXIS_CONSUMED
AXIS_GENERIC
AXIS_SERVO
AXIS_SERVO_DRIVE
AXIS_VIRTUAL</td><td>DINT</td><td>GSV*</td><td>Allows access to all axis status bits in one 32-bit word. This attribute is the same as the Axis Status tag.<table><tr><th>Axis Status</th><th>Bit</th></tr><tr><td>Servo Action Status</td><td>0</td></tr><tr><td>Drive Enable Status</td><td>1</td></tr><tr><td>Shutdown Status</td><td>2</td></tr><tr><td>Config Update In Process</td><td>3</td></tr><tr><td>Inhibit Status</td><td>4</td></tr></table></td></tr>
<tr><td>Axis Type</td><td>AXIS_GENERIC
AXIS_SERVO
AXIS_SERVO_DRIVE</td><td>INT</td><td>GSV
SSV</td><td>The Axis Type attribute is used to establish the intended use of the axis.<table><tr><th>If</th><th>Then set the attribute to</th></tr><tr><td>The axis is unused in the application, which is a common occurrence when there are an odd number of axes in the system</td><td>0</td></tr><tr><td>You only want the position information from the feedback interface</td><td>1</td></tr><tr><td>The axis is intended for full servo operation</td><td>2</td></tr></table>Axis Type is not only used to qualify many operations associated with the axis servo loop, it also controls the behavior of the servo module’s Axis Status indicators. An Axis Type of ‘1’ (Feedback Only) results in the DRIVE LED being blanked, while a value of ‘0’ (Unused) blanks the FDBK and DRIVE status indicators.
This controller attribute is also replicated in the motion module.
Logix Designer application also uses the current configured value for Axis Type to control the look of many of the dialog boxes associated with configuring an axis.</td></tr>
</table>

Attribute	Axis Type	Data Type	Access	Description
Backlash Reversal Offset	AXIS_SERVO AXIS_SERVO_DRIVE	REAL	GSV SSV	Backlash Reversal Offset provides the user the capability to compensate for positional inaccuracy introduced by mechanical backlash. For example, power-train type applications require a high level of accuracy and repeatability during machining operations. Axis motion is often generated by a number of mechanical components such as a motor, a gearbox, and a ball-screw, which can introduce inaccuracies and which are subject to wear over their lifetime. Hence, when an axis is commanded to reverse direction, mechanical play in the machine (through the gearing, ball-screw, and so on.) may result in a small amount of motor motion without axis motion. As a result, the feedback device may indicate movement even though the axis did not move. Compensation for mechanical backlash can be achieved by adding a directional offset, specified by the Backlash Reversal Offset attribute, to the motion planner's command position as it is applied to the associated servo loop. Whenever the commanded velocity changes sign (a reversal), the Logix controller adds, or subtracts, the Backlash Distance value from the current commanded position. This causes the servo to immediately move the motor to the other side of the backlash window and engage the load. It is important to note that the application of this directional offset is completely transparent to the user; the offset does not have any effect on the value of the Command Position attribute. If a value of zero is applied to the Backlash Reversal Offset, the feature is effectively disabled. Once enabled by a non-zero value, and the load is engaged by a reversal of the commanded motion, changing the Backlash Reversal Offset can cause the axis to shift as the offset correction is applied to the command position.
Backlash Stabilization Window	AXIS_SERVO AXIS_SERVO_DRIVE	REAL	GSV SSV	The Backlash Stabilization Window attribute is used to control the Backlash Stabilization feature in the servo control loop. What follows is a description of this feature and the general backlash instability phenomenon. Mechanical backlash is a common problem in applications that utilize mechanical gearboxes. The problem stems from the fact that until the input gear is turned to the point where its proximal tooth contacts an adjacent tooth of the output gear, the reflected inertia of the output is not felt at the motor. In other words, when the gear teeth are not engaged, the system inertia is reduced to the motor inertia. If the servo loop is tuned for peak performance with the load applied, the axis is at best under-damped and at worst unstable in the condition where the gear teeth are not engaged. In the worst case scenario, the motor axis and the input gear oscillates wildly between the limits imposed by the output gear teeth. The net effect is a loud buzzing sound when the axis is at rest. If this situation persists, the gearbox wears out prematurely. To prevent this condition, the conventional approach is to de-tune the servo so that the axis is stable without the gearbox load applied. Unfortunately, system performance suffers.' Due to its non-linear, discontinuous nature, adaptive tuning algorithms generally fall short of addressing the backlash problem. However, a very effective backlash compensation algorithm can be demonstrated using the Torque Scaling gain. The key to this algorithm is the tapered Torque Scaling profile as a function of the position error of the servo loop. The reason for the tapered profile, as opposed to a step profile, is that when the position error exceeds the backlash distance, a step profile would create a very large discontinuity in the torque output. This repulsing torque tends to slam the axis back against the opposite gear tooth and perpetuate the buzzing effect. The tapered Torque Scaling profile is only run when the acceleration command to the servo loop is zero, that is, when we are not commanding any acceleration or deceleration that would engage the teeth of the gearbox. Properly configured with a suitable value for the Backlash Stabilization Window, this algorithm entirely eliminates the gearbox buzz without sacrificing any servo performance. The Backlash Stabilization parameter determines the width of the window over which backlash stabilization is applied. In general, this value should be set to the measured backlash distance. A Backlash Stabilization Window value of zero effectively disables the feature. (Patent Pending)

Attribute	Axis Type	Data Type	Access	Description
Brake Engage Delay Time	AXIS_SERVO_DRIVE	REAL	GSV SSV	Seconds The Brake Engage Delay attribute controls the amount of time that the drive continues to apply torque to the motor after the motor brake output is changed to engage the brake. This gives time for the motor brake to engage. This is the sequence of events associated with engaging the motor brake. • Disable axis is initiated. (via MSF or drive disable fault action) • Drive stops tracking command reference. (Servo Action Status bit clears.) • Decel to zero speed using configured Stopping Torque. • Zero speed or Stopping Time Limit is reached. • Turn motor brake output off to engage the motor brake. • Wait Brake Engage Delay Time. • Disable the drive power structure. (Drive Enable Status bit clears.) If the axis is shutdown through a fault action or motion instruction, the drive power structure is disabled immediately and the motor brake is engaged immediately. • Drive stops tracking command reference. (Servo Action Status bit clears.) • Disable drive power structure, (Drive Enable Status bit clears.) • Turn off brake output to engage brake.
Brake Release Delay Time	AXIS_SERVO_DRIVE	REAL	GSV SSV	Seconds The Brake Release Delay attribute controls the amount of time that the drive holds off tracking command reference changes after the brake output is changed to release the brake. This gives time for the brake to release. This is the sequence of events associated with engaging the brake. • Enable axis is initiated. (via MSO or MAH) • Drive power structure enabled. (Drive Enable Status bit sets.) • Turn motor brake output on to release the brake.** • Wait Brake Release Delay Time. • Track command reference. (Servo_Action_Status bit sets) **The drive does not release the brake unless there is holding torque.
Bus Ready Status	AXIS_SERVO_DRIVE	BOOL	Tag	If the bit is: • ON — The voltage of the drive's dc bus is high enough for operation. • OFF — The voltage of the drive's dc bus is too low.
Bus Regulator Capacity	AXIS_SERVO_DRIVE	REAL	GSV Tag	**Important:** To use this attribute, choose it as one of the attributes for Real Time Axis Information for the axis. Otherwise, you won't see the right value as the axis runs. See Axis Info Select 1. The present utilization of the axis bus regulator as a percent of rated capacity.
Bus Regulator ID	AXIS_SERVO_DRIVE	INT	GSV	The Bus Regulator ID attribute contains the enumeration of the A-B Bus Regulator or System Shunt catalog numbers associated with the axis. If the Bus Regulator ID does not match that of the actual bus regulator or shunt hardware, an error is generated during the drive configuration process.

Attribute	Axis Type	Data Type	Access	Description
C2C Connection Instance	AXIS_CONSUMED AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE AXIS_VIRTUAL	SINT	GSV	Producer/Consumed axis's associated C2C connection instance in reference to the C2C map instance When Axis Data Type is specified to be 'Consumed', then this axis is associated to the consumed data by specifying the C2C Map Instance and the C2C Connection Instance. This attribute is the connection instance under the C2C map instance, which provides the axis data being sent to it from another axis via a C2C connection. For all other Axis Data Types. if this axis is to be produced, then this attribute is set to the connection instance under the local controller's map instance (1) that is used to send the remote axis data via the C2C connection.
C2C Map Instance	AXIS_CONSUMED AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE AXIS_VIRTUAL	SINT	GSV	Producer/Consumed axis's associated C2C map instance When the Axis Data Type attribute is specified to be 'Consumed', then this axis is associated to the consumed data by specifying the C2C Map Instance and the C2C Connection Instance. For all other Axis Data Types, if this axis is to be produced, then this attribute is set to 1 to indicate that the connection is off of the local controller's map instance.
Command Acceleration	AXIS_CONSUMED AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE AXIS_VIRTUAL	REAL	GSV Tag	**Important:** To use this attribute, make sure Auto Tag Update is Enabled for the motion group (default setting). Otherwise, you won't see the right value as the axis runs. Command Acceleration in Position Units / Sec^2 Command Acceleration is the commanded speed of an axis, in the configured axis Position Units per second per second, as generated by any previous motion instructions. It is calculated as the current increment to the command velocity per base update interval. Command Acceleration is a signed value—the sign (+ or -) depends on which direction the axis is being commanded to move. Command Acceleration is a signed floating-point value. Its resolution does not depend on the Averaged Velocity Timebase, but rather on the conversion constant of the axis and the fact that the internal resolution limit on command velocity is 0.00001 feedback counts per base update period per base update period.

Attribute	Axis Type	Data Type	Access	Description
Command Position	AXIS_CONSUMED AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE AXIS_VIRTUAL	REAL	GSV Tag	**Important:** To use this attribute, make sure Auto Tag Update is Enabled for the motion group (default setting). Otherwise, you won't see the right value as the axis runs. Command Position in Position Units Command Position is the desired or commanded position of a physical axis, in the configured Position Units of that axis, as generated by the controller in response to any previous motion Position Control instruction. Command Position data is transferred by the controller to a physical axis as part of an ongoing synchronous data transfer process, which results in a delay of one base update period. Thus, the Command Position value that is obtained is the command position that is acted upon by the physical servo axis one base update period from now. The figure below shows the relationship between Actual Position, Command Position, and Position Error for an axis with an active servo loop. Actual Position is the current position of the axis as measured by the feedback device (for example, encoder). Position error is the difference between the Command and Actual Positions of the servo loop, and is used to drive the motor to make the actual position equal to the command position. Command position is useful when performing motion calculations and incremental moves based on the current position of the axis while the axis is moving. Using command position rather than actual position avoids the introduction of cumulative errors due to the position error of the axis at the time the calculation is performed.
Command Velocity	AXIS_CONSUMED AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE AXIS_VIRTUAL	REAL	GSV Tag	**Important:** To use this attribute, make sure Auto Tag Update is Enabled for the motion group (default setting). Otherwise, you won't see the right value as the axis runs. Command Velocity in Position Units / Seconds Command Velocity is the commanded speed of an axis, in the configured axis Position Units per second, as generated by any previous motion instructions. It is calculated as the current increment to the command position per base update interval. Command Velocity is a signed value—the sign (+ or -) depends on which direction the axis is being commanded to move. Command Velocity is a signed floating-point value. Its resolution does not depend on the Averaged Velocity Timebase, but rather on the conversion constant of the axis and the fact that the internal resolution limit on command velocity is 0.00001 feedback counts per base update.
Common Bus Fault	AXIS_SERVO_DRIVE	BOOL	Tag	The drive shuts down if you give it 3-phase power while it's configured for Common Bus Follower mode. If that happens, this bit turns on.
Commutation Fault	AXIS_SERVO_DRIVE	DINT	BOOL	Set when there is a problem with the commutation feedback source associated with the drive axis that prevents the drive from receiving accurate or reliable motor shaft information to perform commutation.
Config Fault	AXIS_CONSUMED AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE AXIS_VIRTUAL	BOOL	Tag	Set when an update operation targeting an axis configuration attribute of an associated motion module failed. For more information about the Configuration Fault, see the Attribute Error Code and Attribute Error ID attributes associated with the motion module. Do you want this fault to give the controller a major fault? • YES — Set the General Fault Type of the motion group = Major Fault. • NO — You must write code to handle these faults.

Attribute	Axis Type	Data Type	Access	Description
Config Update In Process	AXIS_CONSUMED AXIS_SERVO AXIS_SERVO_DRIVE AXIS_VIRTUAL	BOOL	Tag	When you use an SSV instruction to change an attribute, the controller sends the change to the motion module. If you want to wait until the change is done, monitor the ConfigUpdateInProcess bit of the axis. If the bit is: • ON — The controller is changing the attribute. • OFF — The change is done.
Continuous Torque Limit	AXIS_SERVO_DRIVE	REAL	GSV SSV	%Rated The Torque limit attribute provides a method for controlling the continuous torque limit imposed by the drive's thermal model of the motor. Increasing the Continuous Torque Limit increases the amount of continuous motor torque allowed before the drive folds back the motor current or the drive declares a motor thermal fault. Motors equipped with special cooling options can be configured with a Continuous Torque Limit of greater than 100% rated to attain higher continuous torque output from the motor. Motors operating in high ambient temperature conditions can be configured with a Continuous Torque Limit of less than 100% rated torque to protect the motor from overheating. The Continuous Torque Limit specifies the maximum percentage of the motor's rated current that the drive can command on a continuous or RMS basis. For example, a Continuous Torque Limit of 150% limits the continuous current delivered to the motor to 1.5 times the continuous current rating of the motor.
Control Sync Fault	AXIS_CONSUMED AXIS_SERVO AXIS_SERVO_DRIVE	BOOL	Tag	If this bit is set, the controller lost communication with the motion module and missed several position updates in a row. • The controller can miss up to 4 position updates. After that, the Control Sync Fault bit is set. The motion module may be faulted, or may fault later. • For a consumed axis, this bit means that communication is lost with the producing controller. This bit clears when communication is reestablished.
Controlled By Transform Status	AXIS_CONSUMED AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE AXIS_VIRTUAL	BOOL	Tag	If the bit is: • ON — A transform is moving the axis. • OFF — A transform isn't moving the axis.

<table>
<tr><th>Attribute</th><th>Axis Type</th><th>Data Type</th><th>Access</th><th>Description</th></tr>
<tr><td>Conversion Constant</td><td>AXIS_CONSUMED
AXIS_GENERIC
AXIS_SERVO
AXIS_SERVO_DRIVE
AXIS_VIRTUAL</td><td>REAL</td><td>GSV
SSV</td><td>Counts / Position Unit
Range = 0.1 - 1e^{12}
Axis_Servo Default = 8000
Axis_Servo_Drive Default = 2000000
To allow axis position to be displayed and motion to be programmed in the position units specified by the Position Unit string attribute, a Conversion Constant must be established for each axis. The Conversion Constant, sometimes known as the K constant, allows the Axis Object to convert the axis position units into feedback counts and vice versa. Specifically, K is the number of feedback counts per Position Unit.
The 1756-M02AE encoder based servo module uses 4X encoder feedback decoding (both edges of channel A and B are counted). The count direction is determined from the direction of the edge and the state of the opposite channel. Channel A leads channel B for increasing count. This is the most commonly used decode mode with incremental encoders, since it provides the highest resolution.
For example, suppose this servo axis utilizes a 1000 line encoder in a motor coupled directly to a 5 pitch lead screw (5 turns per inch). With a user defined Position Unit of Inches, the conversion constant is calculated as shown below:
K = 1000 Lines/Rev * 4 Counts/Line * 5 Revs/Inch = 20,000 Counts/Inch.
ATTENTION: If ‘Conversion Constant’ is changed, it invalidates the attributes you can set with ‘Position Unit’ conversions in ‘Description’ column. To be valid, the ‘Conversion Constant’ must be set to the desired value before setting (including defaulting) any of the affected attributes.</td></tr>
<tr><td>Coordinated Motion Status</td><td>AXIS_CONSUMED
AXIS_GENERIC
AXIS_SERVO
AXIS_SERVO_DRIVE
AXIS_VIRTUAL</td><td>DINT</td><td>GSV
Tag</td><td>Set if any coordinated motion profile is currently active upon the axis. It is cleared as soon as Coordinated Motion is complete or stopped.
<table>
<tr><th>Motion Status</th><th>Bit</th><th>Motion Status</th><th>Bit</th></tr>
<tr><td>Accel Status</td><td>0</td><td>Move Pending Status</td><td>8</td></tr>
<tr><td>Decel Status</td><td>1</td><td>Move Pending Queue Full Status</td><td>9</td></tr>
<tr><td>Actual Pos Tolerance Status</td><td>2</td><td>Reserved</td><td>10</td></tr>
<tr><td>Command Pos Tolerance Status</td><td>3</td><td>Reserved</td><td>11</td></tr>
<tr><td>Stopping Status</td><td>4</td><td>Reserved</td><td>12</td></tr>
<tr><td>Reserved</td><td>5</td><td>Coordinate System in a Source CS</td><td>13</td></tr>
<tr><td>Move Status</td><td>6</td><td>Coordinate System in a Target CS</td><td>14</td></tr>
<tr><td>Transition Status</td><td>7</td><td></td><td></td></tr>
</table></td></tr>
<tr><td>Damping Factor</td><td>AXIS_SERVO
AXIS_SERVO_DRIVE</td><td>REAL</td><td>GSV
SSV</td><td>The Damping Factor attribute value is used in calculating the maximum Position Servo Bandwidth (see below) during execution of the MRAT (Motion Run Axis Tune) instruction. In general, the Damping Factor attribute controls the dynamic response of the servo axis. When gains are tuned using a small damping factor (like 0.7), a step response test performed on the axis would demonstrate under-damped behavior with velocity overshoot. A gain set generated using a larger damping factor, like 1.0, produces a system step response that has no overshoot, but has a significantly lower servo bandwidth. The default value for the Damping Factor of 0.8 should work fine for most applications.</td></tr>
<tr><td>DC Bus Voltage</td><td>AXIS_SERVO_DRIVE</td><td>DINT</td><td>GSV
Tag</td><td>**Important:** To use this attribute, choose it as one of the attributes for Real Time Axis Information for the axis. Otherwise, you won’t see the right value as the axis runs. See Axis Info Select 1.
Volts
This parameter is the present voltage on the DC Bus of the drive.</td></tr>
</table>

Attribute	Axis Type	Data Type	Access	Description
Decel Status	AXIS_CONSUMED AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE AXIS_VIRTUAL	BOOL	Tag	Set if the axis is currently being commanded to decelerate. Use the Accel Status bit and the Decel Status bit to see if the axis is accelerating or decelerating. If both bits are off, then the axis is moving at a steady speed or is at rest.
Direct Drive Ramp Rate	AXIS_SERVO	REAL	GSV SSV	Volts/Second The Direct Drive Ramp Rate attribute contains a slew rate for changing the output voltage when the Direct Drive On (MDO) instruction is executed. A Direct Drive Ramp Rate of 0, disables the output ramp rate limiter, allowing the Direct Drive On voltage to be applied directly.
Directional Scaling Ratio	AXIS_SERVO	REAL	GSV SSV	In some cases, the speed or velocity scaling of the external drive actuator may be directionally dependent. This non-linearity can be substantial in hydraulic applications. To compensate for this behavior, the Directional Scaling Ratio attribute can be applied to the Velocity Scaling based on the sign of the Servo Output. Specifically, the Velocity Scaling value is scaled by the Directional Scaling Ratio when the sign of the Servo Output is positive. Thus, the Directional Scaling Ratio is the ratio of the Velocity Scaling in the positive direction (positive servo output) to the Velocity Scaling in the negative direction (negative servo output). The value for the Directional Scaling ratio can be empirically determined by running the auto-tune procedure in the positive direction and then in the negative direction and calculating the ratio of the resulting Velocity/Torque Scaling values.
Drive Axis ID	AXIS_SERVO_DRIVE	INT	GSV	Product Code of Drive Amplifier The Drive ID attribute contains the ASA Product Code of the drive amplifier associated with the axis. If the Product Code does not match that of the actual drive amplifier, an error is generated during the configuration process.
Drive Capacity	AXIS_SERVO_DRIVE	REAL	GSV Tag	**Important:** To use this attribute, choose it as one of the attributes for Real Time Axis Information for the axis. Otherwise, you won't see the right value as the axis runs. See Axis Info Select 1. The present utilization of drive capacity as a percent of rated capacity.
Drive Control Voltage Fault	AXIS_SERVO_DRIVE	BOOL	Tag	Set when the power supply voltages associated with the drive circuitry fall outside of acceptable limits.
Drive Cooling Fault	AXIS_SERVO_DRIVE	BOOL	Tag	Set when the ambient temperature surrounding the drive's control circuitry temperature exceeds the drive ambient shut-down temperature.

<table>
<tr><th>Attribute</th><th>Axis Type</th><th>Data Type</th><th>Access</th><th>Description</th></tr>
<tr><td>Drive Enable Input Fault</td><td>AXIS_SERVO_DRIVE</td><td>BOOL</td><td>Tag</td><td>This fault would be declared if one of two possible conditions occur: 1) If an attempt is made to enable the axis (typically via MSO or MAH instruction) while the Drive Enable Input is inactive. 2) If the Drive Enable Input transitions from active to inactive while the axis is enabled.
This fault can only occur when the Drive Enable Input Fault Handling bit is set in the Fault Configuration Bits attribute.
If the Drive Enable Input Fault Action is set for Stop Command and the axis is stopped as a result of a Drive Enable Input Fault, the faulted axis cannot be moved until the fault is cleared. Any attempt to move the axis in the faulted state using a motion instruction results in an instruction error.
If the Drive Enable Fault Action setting is Status Only or Stop Command and an attempt is made to enable the axis (typically via MSO or MAH instruction) while the Drive Enable Input is active, the axis enables in the faulted state indicating a Drive Enable Input Fault. When the Drive Enable Fault Action setting is Stop Command, instructions that enable the axis and initiate motion (MAH, MRAT, MAHD) abort the motion process leaving the instruction with the IP and PC bits clear.
This fault condition is latched and requires execution of an explicit MAFR (Motion Axis Fault Reset) or MASR (Motion Axis Shutdown Reset) instruction to clear. Attempting to clear the fault while the drive enable input is still inactive and the drive is enabled does not work. However, the drive enable input fault may be cleared with the drive enable input inactive if the drive is disabled.
If the Drive Enable Input Checking bit is clear, then the state of the Drive Enable Input is irrelevant. So, no fault would be declared in any of the above conditions.</td></tr>
<tr><td>Drive Enable Input Fault Action</td><td>AXIS_SERVO_DRIVE</td><td>SINT</td><td>GSV
SSV</td><td><table><tr><th>Fault Action</th><th>Value</th></tr><tr><td>Shutdown</td><td>0</td></tr><tr><td>Disable Drive</td><td>1</td></tr><tr><td>Stop Motion</td><td>2</td></tr><tr><td>Status Only</td><td>3</td></tr></table></td></tr>
<tr><td>Drive Enable Status</td><td>AXIS_CONSUMED
AXIS_GENERIC
AXIS_SERVO
AXIS_SERVO_DRIVE
AXIS_VIRTUAL</td><td>BOOL</td><td>Tag</td><td>AXIS_SERVO
If this bit is:
• ON — The Drive Enable output of the axis is on.
• OFF — The Drive Enable output of the axis is off.
AXIS_SERVO_DRIVE
If this bit is:
• ON — The drive’s power structure is active.
• OFF — The drive’s power structure is not active.
AXIS_VIRTUAL
• Bit is OFF.</td></tr>
<tr><td>Drive Fault</td><td>AXIS_SERVO</td><td>BOOL</td><td>Tag</td><td>If this bit is set, then the external servo drive detected a fault and communicated the existence of this fault to the servo module via the Drive Fault input. This fault condition is latched and requires execution of an explicit MAFR (Motion Axis Fault Reset) or MASR (Motion Axis Shutdown Reset) instruction to clear.</td></tr>
</table>

<table>
<tr><th>Attribute</th><th>Axis Type</th><th>Data Type</th><th>Access</th><th>Description</th></tr>
<tr><td>Drive Fault</td><td>AXIS_SERVO_DRIVE</td><td>DINT</td><td>Tag</td><td>Allows access to all drive fault bits in one 32-bit word. This tag is the same as the Drive Fault Bits attribute.
<table>
<tr><th>Tag</th><th>Bit</th><th>Tag</th><th>Bit</th></tr>
<tr><td>Pos Soft Overtravel Fault</td><td>0</td><td>Overload Fault</td><td>16</td></tr>
<tr><td>Neg Soft Overtravel Fault</td><td>1</td><td>Drive Overtemp Fault</td><td>17</td></tr>
<tr><td>Pos Hard Overtravel Fault</td><td>2</td><td>Motor Overtemp Fault</td><td>18</td></tr>
<tr><td>Neg Hard Overtravel Fault</td><td>3</td><td>Drive Cooling Fault</td><td>19</td></tr>
<tr><td>Mot Feedback Fault</td><td>4</td><td>Drive Control Voltage Fault</td><td>20</td></tr>
<tr><td>Mot Feedback Noise Fault</td><td>5</td><td>Feedback Fault</td><td>21</td></tr>
<tr><td>Aux Feedback Fault</td><td>6</td><td>Commutation Fault</td><td>22</td></tr>
<tr><td>Aux Feedback Noise Fault</td><td>7</td><td>Drive Overcurrent Fault</td><td>23</td></tr>
<tr><td>Reserved</td><td>8</td><td>Drive Overvoltage Fault</td><td>24</td></tr>
<tr><td>Drive Enable Input Fault</td><td>9</td><td>Drive Undervoltage Fault</td><td>25</td></tr>
<tr><td>Common Bus Fault</td><td>10</td><td>Power Phase Loss Fault</td><td>26</td></tr>
<tr><td>Precharge Overload Fault</td><td>11</td><td>Position Error Fault</td><td>27</td></tr>
<tr><td>Guard Fault Exists</td><td>12</td><td>SERCOS Fault</td><td>28</td></tr>
<tr><td>Ground Short Fault</td><td>13</td><td>Overtravel Fault</td><td>29</td></tr>
<tr><td>Drive Hard Fault</td><td>14</td><td>Reserved</td><td>30</td></tr>
<tr><td>Overspeed Fault</td><td>15</td><td>Manufacturer Specific Fault</td><td>31</td></tr>
</table>
Do you want any of these faults to give the controller a major fault?
• YES - Set the General Fault Type of the motion group = Major Fault.
• NO - You must write code to handle these faults.</td></tr>
<tr><td>Drive Fault Action</td><td>AXIS_SERVO</td><td>SINT</td><td>GSV
SSV</td><td>
<table>
<tr><th>Fault Action</th><th>Value</th></tr>
<tr><td>Shutdown</td><td>0</td></tr>
<tr><td>Disable Drive</td><td>1</td></tr>
<tr><td>Stop Motion</td><td>2</td></tr>
<tr><td>Status Only</td><td>3</td></tr>
</table>
This controller attribute is also replicated in the motion module.</td></tr>
</table>

<table>
<tr><th>Attribute</th><th>Axis Type</th><th>Data Type</th><th>Access</th><th>Description</th></tr>
<tr><td>Drive Fault Bits</td><td>AXIS_SERVO_DRIVE</td><td>DINT</td><td>GSV</td><td>Allows access to all drive fault bits in one 32-bit word. This attribute is the same as the Drive Fault tag.
<table>
<tr><th>Tag</th><th>Bit</th><th>Tag</th><th>Bit</th></tr>
<tr><td>Pos Soft Overtravel Fault</td><td>0</td><td>Overload Fault</td><td>16</td></tr>
<tr><td>Neg Soft Overtravel Fault</td><td>1</td><td>Drive Overtemp Fault</td><td>17</td></tr>
<tr><td>Pos Hard Overtravel Fault</td><td>2</td><td>Motor Overtemp Fault</td><td>18</td></tr>
<tr><td>Neg Hard Overtravel Fault</td><td>3</td><td>Drive Cooling Fault</td><td>19</td></tr>
<tr><td>Mot Feedback Fault</td><td>4</td><td>Drive Control Voltage Fault</td><td>20</td></tr>
<tr><td>Mot Feedback Noise Fault</td><td>5</td><td>Feedback Fault</td><td>21</td></tr>
<tr><td>Aux Feedback Fault</td><td>6</td><td>Commutation Fault</td><td>22</td></tr>
<tr><td>Aux Feedback Noise Fault</td><td>7</td><td>Drive Overcurrent Fault</td><td>23</td></tr>
<tr><td>Reserved</td><td>8</td><td>Drive Overvoltage Fault</td><td>24</td></tr>
<tr><td>Drive Enable Input Fault</td><td>9</td><td>Drive Undervoltage Fault</td><td>25</td></tr>
<tr><td>Common Bus Fault</td><td>10</td><td>Power Phase Loss Fault</td><td>26</td></tr>
<tr><td>Precharge Overload Fault</td><td>11</td><td>Position Error Fault</td><td>27</td></tr>
<tr><td>GuardFaultExists</td><td>12</td><td>SERCOS Fault</td><td>28</td></tr>
<tr><td>Ground Short Fault</td><td>13</td><td>Overtravel Fault</td><td>29</td></tr>
<tr><td>Drive Hard Fault</td><td>14</td><td>Reserved</td><td>30</td></tr>
<tr><td>Overspeed Fault</td><td>15</td><td>Manufacturer Specific Fault</td><td>31</td></tr>
</table>
Do you want any of these faults to give the controller a major fault?
• YES — Set the General Fault Type of the motion group = Major Fault.
• NO — You must write code to handle these faults.</td></tr>
<tr><td>Drive Fault Input Status</td><td>AXIS_SERVO</td><td>BOOL</td><td>Tag</td><td>Digital output from the drive that shows if there is a fault.
If this bit is:
• ON — The drive faulted.
• OFF — The drive does not have a fault.</td></tr>
<tr><td>Drive Hard Fault</td><td>AXIS_SERVO_DRIVE</td><td>BOOL</td><td>Tag</td><td>Set when the drive detects a serious hardware fault.</td></tr>
<tr><td>Drive Model Time Constant</td><td>AXIS_SERVO
AXIS_SERVO_DRIVE</td><td>REAL</td><td>GSV
SSV</td><td>Seconds
The value for the Drive Model Time Constant represents the lumped model time constant for the drive's current loop used by the MRAT instruction to calculate the Maximum Velocity and Position Servo Bandwidth values. The Drive Model Time Constant is the sum of the drive's current loop time constant, the feedback sample period, and the time constant associated with the velocity feedback filter. This value is set to a default value when you configure the axis.
<table>
<tr><th>For this Axis type</th><th>Details</th></tr>
<tr><td>AXIS_SERVO</td><td>This value is only used by MRAT when the axis is configure for an External Torque Servo Drive..</td></tr>
<tr><td>AXIS_SERVO_DRIVE</td><td>Since the bandwidth of the velocity feedback filter is determined by the resolution of the feedback device, the value for the Drive Model Time Constant is smaller when hi resolution feedback devices are selected.</td></tr>
</table></td></tr>
</table>

Attribute	Axis Type	Data Type	Access	Description
Drive Overcurrent Fault	AXIS_SERVO_DRIVE	BOOL	Tag	Set when drive output current exceeds the predefined operating limits for the drive.
Drive Overtemp Fault	AXIS_SERVO_DRIVE	BOOL	Tag	Set when the drive's temperature exceeds the drive shutdown temperature.
Drive Overvoltage Fault	AXIS_SERVO_DRIVE	BOOL	Tag	Set when drive DC bus voltage exceeds the predefined operating limits for the bus.
Drive Polarity	AXIS_SERVO_DRIVE	DINT	GSV SSV	0 = Custom Polarity 1 = Positive Polarity 2 = Negative Polarity **Custom Polarity** Custom Polarity is used to enable custom polarity configurations using the various polarity parameters defined by the SERCOS Interface standard. **Positive/Negative Polarity** Positive and Negative Polarity bit attribute determines the overall polarity of the servo loop of the drive. The advanced polarity parameters are automatically set based on whether the Drive Polarity is configured as Positive or Negative. Proper wiring guarantees that the servo loop is closed with negative feedback. However, there is no such guarantee that the servo drive has the same sense of forward direction as the user for a given application. Negative Polarity inverts the polarity of the command position and actual position data of the servo drive. Thus, selecting Positive or Negative Drive Polarity makes it possible to configure the positive direction sense of the drive to agree with that of the user. This attribute is configured automatically using the MRHD and MAHD motion instructions. See the Logix Motion Instruction Specification for more information on these hookup diagnostic instructions.

Attribute	Axis Type	Data Type	Access	Description
Drive Resolution	AXIS_SERVO_DRIVE	DINT	GSV	Drive Counts / Drive Unit The Drive Resolution attribute determines how many Drive Counts there are in a Drive Unit. Drive Units may be configured as Revs, Inches, or Millimeters depending on the drive application. Furthermore, the configured Drive Unit may apply to a motor or auxiliary feedback device. All position, velocity, and acceleration data to the drive is scaled from the user's Position Units to Drive Units based on the Drive Resolution and Conversion Constant. The ratio of the Conversion Constant to Drive Resolution determines the number of Position Units in a Drive Unit. Conversion Constant / Drive Resolution = Drive Units (rev, inch, or mm) / Position Unit Conversely, all position, velocity, and acceleration data from the drive is scaled from the user's Position Units to Drive Units based on the Drive Resolution and Conversion Constant. The ratio of Drive Resolution and the Conversion Constant determines the number of Position Units in a Drive Unit. Drive Resolution / Conversion Constant = Position Units / Drive Unit (rev, inch, or mm) In general, the Drive Resolution value may be left at its default value of 200000 Drive Counts per Drive Unit, independent of the resolution of the feedback device(s) used by the drive. This is because the drive has its own set of scale factors that it uses to relate feedback counts to drive counts. **Drive Travel Range Limit** Because the drive's position parameters are ultimately limited to signed 32-bit representation per the SERCOS standard, the Drive Resolution parameter impacts the drive's travel range. The equation for determining the maximum travel range based on Drive Resolution is as follows: Drive Travel Range Limit = +/- 2,147,483,647 / Drive Resolution. Based on a default value of 200,000 Drive Counts per Drive Unit, the drive's range limit is 10,737 Drive Units. While it is relatively rare for this travel range limitation to present a problem, it is a simple matter to lower the Drive Resolution to increase the travel range. The downside of doing so is that the position data is then passed with lower resolution that could affect the smoothness of motion. **Fractional Unwind** In some cases, however, the user may also want to specifically configure Drive Resolution value to handle fractional unwind applications or multi-turn absolute applications requiring cyclic compensation. In these cases, where the Unwind value for a rotary application does not result in an integer value, the Rotational Position Scaling attribute may be modified to a value that is integer divisible by the Unwind value. The following examples demonstrate how the Drive Resolution value may be used with the Conversion Constant to handle various applications.

Attribute	Axis Type	Data Type	Access	Description
Drive Resolution (continued)				**Rotary Gear-Head WITHOUT Aux Feedback Device** Based on a rotary motor selection, Drive Resolution would be expressed as Drive Counts per Motor Rev and be applied to the Rotational Position Resolution IDN. You would set the Conversion Constant to Drive Counts per user-defined Position Unit. If it is a 3:1 gearbox, and the user's Position Unit is Revs of the gear output shaft, the Conversion Constant is 200,000/3, which is irrational! But, in this case, you could simply set the Drive Resolution to 300,000 Drive Counts/Motor Rev and the Conversion Constant could then be set to 100,000 Drive Counts/Output Shaft Rev. This system would work with this configuration without any loss of mechanical precision, that is, a move of 1 output shaft revolution would move the output shaft 1 revolution. **Linear Ball-Screw WITHOUT Aux Feedback Device** Based on a rotary motor selection, Drive Resolution would be expressed as Drive Counts per Motor Rev and be applied to the Rotational Position Resolution IDN. You would set the Conversion Constant to Drive Counts per user-defined Position Unit. If it is a 5mm pitch ball-screw, and the user's Position Unit is mm, you set the Conversion Constant to 200,000/5 or 40,000 Drive Counts per mm based on the default Drive Resolution value of 200,000 Drive Counts/Motor Rev. If the pitch is irrational, the method for addressing this is the same as described in Rotary Gear-Head WITHOUT Aux Feedback Device. **Rotary Gear-Head WITH Aux Feedback Device** Based on a rotary motor feedback selection, Drive Resolution would be expressed as Drive Counts per Aux Rev and be applied to the Rotational Position Resolution IDN. Now that position is based on the auxiliary feedback device according to the Servo Loop Configuration, the Data Reference bit of the various Scaling Types should be Load Referenced rather than Motor Referenced. The motor feedback would be rotary and resolution expressed in cycles per motor rev. The aux feedback device is also rotary and its resolution expressed in cycles per aux rev. The Aux Feedback Ratio would be set to the number of aux feedback revs per motor rev and internally applied to IDNs 121 and 122 for the purpose of relating position servo loop counts to velocity servo loop counts in a dual servo loop configuration. The Aux Feedback Ratio attribute is also used in range limit and default value calculations during configuration based on the selected motor’s specifications. If the application uses a 3:1 gearbox, and the user's Position Unit is Revs of the gearbox output shaft, the Conversion Constant is still rational, since our scaling is Load Referenced! You set the Conversion Constant to 200,000 Drive Counts/Output Shaft Rev based on the default Drive Resolution value of 200,000 Drive Counts/Aux Rev. The system would work in this configuration without any loss of mechanical precision, that is, a move of 1 output shaft revolution would move the output shaft 1 revolution.

Attribute	Axis Type	Data Type	Access	Description
Drive Resolution (continued)				**Linear Ball-Screw/Ball-Screw Combination WITH Aux Feedback Device** Based on a linear aux feedback selection, Drive Resolution would be expressed as Drive Counts per Linear Unit, say Millimeters (Metric bit selection), and be applied to the Linear Position Data Scaling IDNs. Now that position is based on the auxiliary feedback device according to the Servo Loop Configuration, the Data Reference bit of the various Scaling Types should again be Load Referenced rather than Motor Referenced. The motor feedback would be rotary and resolution expressed in cycles per motor rev. The aux feedback device is now linear and its resolution expressed in cycles per, say, mm. The Aux Feedback Ratio would be set to the number of aux feedback units (mm) per motor rev and internally applied to IDN 123 to relate position servo loop counts to velocity servo loop counts in a dual servo loop configuration. The Aux Feedback Ratio attribute is also used in range limit and default value calculations during configuration based on the selected motor's specifications. If the application uses a 3:1 gearbox and a 5 mm pitch ball-screw, and your Position Unit is cm, the Conversion Constant is again rational, since we are Load Referenced! You set the Conversion Constant to 20,000 Drive Counts/cm based on the default Drive Resolution value of 200000 Drive Counts/mm. This system would work in this configuration without any loss of mechanical precision, that is, a move of 10 cm would move the actuator 10 cm.

<table>
<tr>
<td>Drive Scaling Bits</td>
<td>AXIS_SERVO_DRIVE</td>
<td>DINT</td>
<td>GSV</td>
<td>
The Drive Scaling Bits attribute configuration is derived directly from the Drive Units attribute.
Bits
0 = Scaling type
0 – standard
1 – custom
1 = Scaling unit
0 – rotary
1 – linear
2 = Linear scaling unit
0 – metric
1 – english
3 = Data Reference
0 – motor
1 – load
Scaling Type
The Scaling Type bit attribute is used to enable custom scaling using the position, velocity, acceleration, and torque scaling parameters defined by the SERCOS Interface standard. When the bit is clear (default), these scaling parameters are all set based on the preferred Rockwell Automation SERCOS drive scaling factors. Currently, there is no Logix support for custom scaling.
Scaling Unit
The Scaling Unit attribute is used to determine whether the controller scales position, velocity, and acceleration attributes based on rotary or linear scaling parameters and their associated Drive Units are defined by the SERCOS Interface standard. When the bit is clear (default), the corresponding bits in the SERCOS Position Data Scaling, Velocity Data Scaling, and Acceleration Data Scaling parameters are also cleared, which instructs the drive to use the rotary scaling parameters. When the bit is set, the corresponding bits in the SERCOS Position Data Scaling, Velocity Data Scaling, and Acceleration Data Scaling parameters are also set, which instructs the drive to use the linear scaling parameters.
Linear Scaling Unit
When the Scaling Unit is set to linear, the Linear Scaling bit attribute is used to determine whether the controller scales position, velocity, and acceleration attributes based on Metric or English Drive Units as defined by the SERCOS Interface standard. When the bit is clear (default), the corresponding bits in the SERCOS Position Data Scaling, Velocity Data Scaling, and Acceleration Data Scaling parameters are also cleared, which instructs the drive to use the Metric scaling parameters. When the bit is set, the corresponding bits in the SERCOS Position Data Scaling, Velocity Data Scaling, and Acceleration Data Scaling parameters are also set, which instructs the drive to scale in English units.
The Linear Scaling Unit bit does not apply if the Scaling Unit is set to rotary.
When interfacing to Rockwell SERCOS drive products, the Standard Drive Units based on the Scaling Unit and Linear Scaling Unit bit selections are shown in the following table:
<table>
<tr><th colspan="3">Standard Drive Units</th></tr>
<tr><th></th><th>Metric</th><th>English</th></tr>
<tr><td>Rotary</td><td>Rev</td><td>Rev</td></tr>
<tr><td>Linear</td><td>Millimeter</td><td>Inch</td></tr>
</table>
Data Reference
The Data Reference bit determines which side of the mechanical transmission to reference position, velocity, acceleration, and torque data. If motor is selected, then position, velocity, acceleration, and torque data is referenced to the motor side of the transmission. If load is selected, then position, velocity, acceleration, and torque data is referenced to the load-side of the transmission. This is only applicable when using an auxiliary feedback device.
</td>
</tr>
</table>

<table>
<tr><th>Attribute</th><th>Axis Type</th><th>Data Type</th><th>Access</th><th>Description</th></tr>
<tr><td>Drive Status Bits</td><td>AXIS_SERVO_DRIVE</td><td>DINT</td><td>GSV</td><td>Allows access to all drive status bits in one 32-bit word. This attribute is the same as the Drive Status tag.
<table>
<tr><th>Tag</th><th>Bit</th><th>Tag</th><th>Bit</th></tr>
<tr><td>Servo Action Status</td><td>0</td><td>Reserved</td><td>14</td></tr>
<tr><td>Drive Enable Status</td><td>1</td><td>Reserved</td><td>15</td></tr>
<tr><td>Shutdown Status</td><td>2</td><td>Velocity Lock Status</td><td>16</td></tr>
<tr><td>Process Status</td><td>3</td><td>Velocity Standstill Status</td><td>17</td></tr>
<tr><td>Bus Ready Status</td><td>4</td><td>Velocity Threshold Status</td><td>18</td></tr>
<tr><td>Reserved</td><td>5</td><td>Torque Threshold Status</td><td>19</td></tr>
<tr><td>Home Input Status</td><td>6</td><td>Torque Limit Status</td><td>20</td></tr>
<tr><td>Reg 1 Input Status</td><td>7</td><td>Velocity Limit Status</td><td>21</td></tr>
<tr><td>Reg 2 Input Status</td><td>8</td><td>Position Lock Status</td><td>22</td></tr>
<tr><td>Pos Overtravel Input Status</td><td>9</td><td>Power Limit Status</td><td>23</td></tr>
<tr><td>Neg Overtravel Input Status</td><td>10</td><td>Reserved</td><td>24</td></tr>
<tr><td>Enable Input Status</td><td>11</td><td>Low Velocity Threshold Status</td><td>25</td></tr>
<tr><td>Accel Limit Status</td><td>12</td><td>High Velocity Threshold Status</td><td>26</td></tr>
<tr><td>Absolute Reference Status</td><td>13</td><td></td><td></td></tr>
</table></td></tr>
<tr><td>Drive Status</td><td>AXIS_SERVO_DRIVE</td><td>DINT</td><td>Tag</td><td>Allows access to all drive status bits in one 32-bit word. This tag is the same as the Drive Status Bits attribute.
<table>
<tr><th>Tag</th><th>Bit</th><th>Tag</th><th>Bit</th></tr>
<tr><td>Servo Action Status</td><td>0</td><td>Reserved</td><td>14</td></tr>
<tr><td>Drive Enable Status</td><td>1</td><td>Reserved</td><td>15</td></tr>
<tr><td>Shutdown Status</td><td>2</td><td>Velocity Lock Status</td><td>16</td></tr>
<tr><td>Process Status</td><td>3</td><td>Velocity Standstill Status</td><td>17</td></tr>
<tr><td>Bus Ready Status</td><td>4</td><td>Velocity Threshold Status</td><td>18</td></tr>
<tr><td>Reserved</td><td>5</td><td>Torque Threshold Status</td><td>19</td></tr>
<tr><td>Home Input Status</td><td>6</td><td>Torque Limit Status</td><td>20</td></tr>
<tr><td>Reg 1 Input Status</td><td>7</td><td>Velocity Limit Status</td><td>21</td></tr>
<tr><td>Reg 2 Input Status</td><td>8</td><td>Position Lock Status</td><td>22</td></tr>
<tr><td>Pos Overtravel Input Status</td><td>9</td><td>Power Limit Status</td><td>23</td></tr>
<tr><td>Neg Overtravel Input Status</td><td>10</td><td>Reserved</td><td>24</td></tr>
<tr><td>Enable Input Status</td><td>11</td><td>Low Velocity Threshold Status</td><td>25</td></tr>
<tr><td>Accel Limit Status</td><td>12</td><td>High Velocity Threshold Status</td><td>26</td></tr>
<tr><td>Absolute Reference Status</td><td>13</td><td></td><td></td></tr>
</table></td></tr>
</table>

<table>
<tr><th>Attribute</th><th>Axis Type</th><th>Data Type</th><th>Access</th><th>Description</th></tr>
<tr><td>Drive Status</td><td>AXIS_SERVO_DRIVE</td><td>BOOL</td><td>Tag</td><td>
<table>
<tr><th>Tag</th><th>Bit</th></tr>
<tr><td>Servo Action Status</td><td>0</td></tr>
<tr><td>Drive Enable Status</td><td>1</td></tr>
<tr><td>Shutdown Status</td><td>2</td></tr>
<tr><td>Process Status</td><td>3</td></tr>
<tr><td>Bus Ready Status</td><td>4</td></tr>
<tr><td>Reserved</td><td>5</td></tr>
<tr><td>Home Input Status</td><td>6</td></tr>
<tr><td>Reg 1 Input Status</td><td>7</td></tr>
<tr><td>Reg 2 Input Status</td><td>8</td></tr>
<tr><td>Pos Overtravel Input Status</td><td>9</td></tr>
<tr><td>Neg Overtravel Input Status</td><td>10</td></tr>
<tr><td>Enable Input Status</td><td>11</td></tr>
<tr><td>Accel Limit Status</td><td>12</td></tr>
<tr><td>Absolute Reference Status</td><td>13</td></tr>
<tr><td>Safe-Off Mode Active Status</td><td>14 (requires Drive firmware revision 1.85 or higher)</td></tr>
</table>
</td></tr>
<tr><td>Drive Thermal Fault Action</td><td>AXIS_SERVO_DRIVE</td><td>SINT</td><td>GSV
SSV</td><td>
<table>
<tr><th>Fault Action</th><th>Value</th></tr>
<tr><td>Shutdown</td><td>0</td></tr>
<tr><td>Disable Drive</td><td>1</td></tr>
<tr><td>Stop Motion</td><td>2</td></tr>
<tr><td>Status Only</td><td>3</td></tr>
</table>
</td></tr>
<tr><td>Drive Undervoltage Fault</td><td>AXIS_SERVO_DRIVE</td><td>BOOL</td><td>Tag</td><td>Set when drive DC bus voltage is below the predefined operating limits for the bus.</td></tr>
<tr><td>Drive Unit</td><td>AXIS_SERVO_DRIVE</td><td>INT</td><td>GSV</td><td>The Drive Unit attribute establishes the unit of measure that is applied to the Drive Resolution attribute value. Units appearing in the enumerated list may be linear or rotary, English or metric. Further discrimination is provided in the enumerated list to specify whether the Drive Unit is referenced directly to the motor or to the external, or auxiliary, feedback.
0 = motor revs
1 = aux revs
2 = motor inches
3 = aux inches
4 = motor mm
5 = aux mm</td></tr>
</table>

<table>
<tr><th>Attribute</th><th>Axis Type</th><th>Data Type</th><th>Access</th><th>Description</th></tr>
<tr><td>Drive Warning Bits</td><td>AXIS_SERVO_DRIVE</td><td>DINT</td><td>GSV*</td><td>
<table>
<tr><th>Warning</th><th>Bit</th></tr>
<tr><td>Drive Overload Warning</td><td>0</td></tr>
<tr><td>Drive Overtemperature Warning</td><td>1</td></tr>
<tr><td>Motor Overtemperature Warning</td><td>2</td></tr>
<tr><td>Cooling Error Warning</td><td>3</td></tr>
</table>
Drive Overload Warning
When the load limit of the motor is exceeded, the Overload Warning bit is set. If the condition persists, an Overload Fault occurs. This warning bit gives the control program an opportunity to reduce motor loading to avoid a future shutdown situation.
Drive Overtemperature Warning
When the over-temperature limit of the drive is exceeded, the Drive Overtemperature Warning bit is set. If the condition persists, a Drive Overtemperature Fault occurs. This warning bit gives the control program an opportunity to reduce motor loading, or increase drive cooling, to avoid a future shutdown situation.
Motor Overtemperature Warning
When the over-temperature limit of the motor is exceeded, the Motor Overtemperature Warning bit is set. If the condition persists, a Motor Overtemperature Fault occurs. This warning bit gives the control program an opportunity to reduce motor loading, or increase motor cooling, to avoid a future shutdown situation.
Cooling Error Warning
When the ambient temperature limit inside the drive enclosure is exceeded, the Cooling Error Warning bit sets. If the condition persists, a Cooling Error Fault occurs. This warning bit gives the control program an opportunity to increase drive cooling to avoid a future shutdown situation.
</td></tr>
</table>

<table>
<tr><th>Attribute</th><th>Axis Type</th><th>Data Type</th><th>Access</th><th>Description</th></tr>
<tr><td>Dynamics Configuration Bits</td><td>AXIS_CONSUMED
AXIS_GENERIC
AXIS_SERVO
AXIS_SERVO_DRIVE
AXIS_VIRTUAL</td><td>DINT</td><td>GSV
SSV</td><td>Revision 16 improved how the controller handles changes to an S-curve profile.
Do you want to return to revision 15 or earlier behavior for S-curves?
• NO — Leave these bits ON (default).
• YES — Turn OFF one or more of these bits.
<table><tr><th>To turn off this change</th><th>Turn o this bi</th></tr><tr><td>Reduced S-curve Stop Delay
This change applies to the Motion Axis Stop (MAS) instruction. It lets you use a higher deceleration jerk to stop an accelerating axis more quickly.
The controller uses the deceleration jerk of the stopping instruction if it is more than the current acceleration jerk.</td><td>0</td></tr><tr><td>Reduced S-curve Velocity Reversals
Before revision 16, you could cause an axis to momentarily reverse direction if you decreased the deceleration jerk while the axis was decelerating. This typically happened if you tried to restart a jog or move with a lower deceleration rate while the axis was stopping. This change prevents the axis from reversing in those situations.</td><td>1</td></tr><tr><td>Reduced S-curve Velocity Overshoots
You can cause an axis to overshoot its programmed speed if you decrease the acceleration jerk while the axis is accelerating. This change keeps to overshoot to no more than 50% of the programmed speed.</td><td>2</td></tr></table></td></tr>
<tr><td>Enable Input Status</td><td>AXIS_SERVO_DRIVE</td><td>BOOL</td><td>Tag</td><td>If this bit is:
• ON — The Enable input is active.
• OFF — The Enable input is inactive.</td></tr>
<tr><td>External Drive Type</td><td>AXIS_SERVO_DRIVE</td><td>DINT</td><td>GSV
SSV</td><td>0 = torque servo
1 = velocity servo
2 = hydraulic servo
When the application requires the servo module axis to interface with an external velocity servo drive, the External Drive Type should be configured for velocity servo. This disables the servo module's internal digital velocity loop. If the External Drive Type attribute is set to torque servo, the servo module's internal digital velocity loop is active. This configuration is the required configuration for interfacing to a torque loop servo drive. If the External Drive Type attribute is set to hydraulic servo, the object enables certain hydraulic servo application features. In general, selecting the hydraulic External Drive Type configures the servo loop the same as selecting the velocity servo External Drive Type.</td></tr>
</table>

<table>
<tr><th>Attribute</th><th>Axis Type</th><th>Data Type</th><th>Access</th><th>Description</th></tr>
<tr><td>Fault Configuration Bits</td><td>AXIS_SERVO
AXIS_SERVO_DRIVE</td><td>DINT</td><td>GSV
SSV</td><td>
<table>
<tr><th>Axis Type</th><th>Fault Configuration</th><th>Bit</th></tr>
<tr><td>AXIS_SERVO</td><td>Soft Overtravel Checking</td><td>0</td></tr>
<tr><td></td><td>Reserved</td><td>1</td></tr>
<tr><td></td><td>Drive Fault Checking</td><td>2</td></tr>
<tr><td></td><td>Drive Fault Normally Closed</td><td>3</td></tr>
<tr><td>AXIS_SERVO_DRIVE</td><td>Soft Overtravel Checking</td><td>0</td></tr>
<tr><td></td><td>Hard Overtravel Checking</td><td>1</td></tr>
<tr><td></td><td>Reserved</td><td>2</td></tr>
<tr><td></td><td>Reserved</td><td>3</td></tr>
<tr><td></td><td>Drive Enable Input Fault Handling</td><td>4</td></tr>
<tr><td></td><td>Drive Enable Input Checking</td><td>5</td></tr>
</table>
Change to rotary or Overtravel Checking requires Home range checks.

Soft Overtravel Checking
Soft overtravel checking is only available for a linear axis.
Do you want a Positive Soft Overtravel Fault or Negative Soft Overtravel Fault to happen if the axis goes outside the configured travel limits?
• YES — Set this bit.
• NO — Clear this bit.
The Maximum Positive Travel and Maximum Negative Travel attributes set the travel limits. This check supplements but does not replace hardware overtravel fault protection that uses hardware limit switches to directly stop axis motion at the drive and deactivate power to the system.

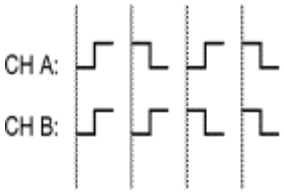
Hard Overtravel Checking
Hard overtravel checking is only available for a linear axis.
Do you want a Positive Hard Overtravel Fault or Negative Hard Overtravel Fault to happen if the axis activates the positive or negative overtravel limit switch inputs?
• YES — Set this bit.
• NO — Clear this bit.

Drive Fault Checking
The motion module provides a dedicated drive fault input for each axis. These inputs may be connected to fault outputs on the external drive (if provided) to notify the servo module of a fault in the drive itself. Set the Drive Fault Checking bit if you are using the servo module's drive fault input, and then specify the drive fault contact configuration of the amplifier's drive fault output as described below.</td></tr>
</table>

Attribute	Axis Type	Data Type	Access	Description
Fault Configuration Bits (continued)		DINT	GSV SSV	**Drive Fault Normally Closed** The Drive Fault Normally Closed bit attribute controls the sense of the Drive Fault input to the servo module. If this bit is set (true), then during normal (fault-free) operation of the drive, the Drive Fault input should be active, that is, 24 Volts. If a drive fault occurs, the drive opens its drive fault output contacts and remove 24 Volts from the servo module's Drive Fault input generating an axis Drive Fault condition. This is the default 'fail-safe' configuration. In some cases, it may be necessary to clear the Drive Fault Normally Closed bit to interface with a drive system that closes its contacts when faulted. This is generally not recommended for 'fail-safe' operation. **Drive Enable Input Fault Handling** When the Drive Enable Input Fault Handling bit is set, it lets the drive post a fault based on the condition of the Drive Enable Input. If an attempt is made to enable the drive axis without an active Drive Enable Input, the drive sets a Drive Enable Input Fault. If the Drive Enable Input ever goes from active to inactive while the drive axis is enabled, the drive also sets a Drive Enable Input Fault. If the Drive Enable Input Fault Handling bit is clear (default), and then the drive does not generate a Drive Enable Input Fault. **Drive Enable Input Checking** When the Drive Enable Input Checking bit is set (the default), the drive regularly checks the current state of the Drive Enable Input. This dedicated input serves as a permissive to enable the drive's power structure and servo loop. Once the drive is enabled, a transition of the Drive Enable Input from active to inactive results in a drive initiated axis stop where the axis is decelerated to a stop using the configured Stopping Torque and then disabled. If the drive enable Input Checking bit is clear, then no Drive Enable Input checking is done, hence, the state of the input is irrelevant to drive operation. The state of the switch is still reported as part of the Drive Status bits attribute.
Feedback Fault	AXIS_SERVO AXIS_SERVO_DRIVE	BOOL	Tag	**AXIS_SERVO** Set for a feedback source when one of the following conditions occurs: • The differential electrical signals for one or more of the feedback channels (for example, A+ and A-, B+ and B-, or Z+ and Z-) are at the same level (high or low). Under normal operation, the differential signals are at opposite levels. The most common cause of this situation is a broken wire between the feedback transducer and the servo module or drive; • Loss of feedback 'power' or feedback 'common' electrical connection between the servo module or drive and the feedback device. The controller latches this fault. Use a Motion Axis Fault Reset (MAFR) or Motion Axis Shutdown Reset (MASR) instruction to clear the fault. **AXIS_SERVO_DRIVE** Set when an issue with one of the feedback sources associated with the drive axis prevents the drive from receiving accurate or reliable position information from the feedback device. Set when one of the feedback sources for the axis cannot send accurate or reliable position information because there is a problem. For AXIS_SERVO axis, possible problems are: • The differential electrical signals for one or more of the feedback channels (for example, A+ and A-, B+ and B-, or Z+ and Z-) are at the same level (high or low). Under normal operation, the differential signals are at opposite levels. The most common cause of this situation is a broken wire between the feedback transducer and the servo module or drive; • Loss of feedback power or common electrical connection between the servo module or drive and the feedback device. The controller latches this fault. Use a Motion Axis Fault Reset (MAFR) or Motion Axis Shutdown Reset (MASR) instruction to clear the fault.

<table>
<tr><th>Attribute</th><th>Axis Type</th><th>Data Type</th><th>Access</th><th>Description</th></tr>
<tr><td>Feedback Fault Action</td><td>AXIS_SERVO
AXIS_SERVO_DRIVE</td><td>SINT</td><td>GSV
SSV</td><td>
<table>
<tr><th>Axis Type</th><th>Fault Configuration</th><th>Bit</th></tr>
<tr><td>AXIS_SERVO</td><td>Soft Overtravel Checking</td><td>0</td></tr>
<tr><td></td><td>Reserved</td><td>1</td></tr>
<tr><td></td><td>Drive Fault Checking</td><td>2</td></tr>
<tr><td></td><td>Drive Fault Normally Closed</td><td>3</td></tr>
<tr><td>AXIS_SERVO_DRIVE</td><td>Soft Overtravel Checking</td><td>0</td></tr>
<tr><td></td><td>Hard Overtravel Checking</td><td>1</td></tr>
<tr><td></td><td>Reserved</td><td>2</td></tr>
<tr><td></td><td>Reserved</td><td>3</td></tr>
<tr><td></td><td>Drive Enable Input Fault Handling</td><td>4</td></tr>
<tr><td></td><td>Drive Enable Input Checking</td><td>5</td></tr>
</table>
This controller attribute is also replicated in the motion module.
</td></tr>
<tr><td>Feedback Noise Fault</td><td>AXIS_SERVO</td><td>BOOL</td><td>Tag</td><td>
Set when there is noise on the feedback device's signal lines.

For example, simultaneous transitions of the feedback A and B channels of an A Quad B is referred to generally as feedback noise.
Feedback noise (shown below) is most often caused by loss of quadrature in the feedback device itself or radiated common-mode noise signals being picked up by the feedback device wiring. You can see these on an oscilloscope.



To troubleshoot the loss of channel quadrature, look for:

physical misalignment of the feedback transducer components
excessive capacitance (or other delays) on the encoder signals
Proper grounding and shielding usually cures radiated noise problems.

The controller latches this fault. Use a Motion Axis Fault Reset (MAFR) or Motion Axis Shutdown Reset (MASR) instruction to clear the fault.
</td></tr>
<tr><td>Feedback Noise Fault Action</td><td>AXIS_SERVO
AXIS_SERVO_DRIVE</td><td>SINT</td><td>GSV
SSV</td><td>
<table>
<tr><th>Fault Action</th><th>Value</th></tr>
<tr><td>Shutdown</td><td>0</td></tr>
<tr><td>Disable Drive</td><td>1</td></tr>
<tr><td>Stop Motion</td><td>2</td></tr>
<tr><td>Status Only</td><td>3</td></tr>
</table>
This controller attribute is also replicated in the motion module.
</td></tr>
</table>

<table>
<tr><th>Attribute</th><th>Axis Type</th><th>Data Type</th><th>Access</th><th>Description</th></tr>
<tr><td>Backlash Compensation</td><td>AXIS_SERVO
AXIS_SERVO_DRIVE</td><td>REAL</td><td>GSV
SSV</td><td>0…100%
It is not unusual for an axis to have enough static friction (sticktion) that even with a significant position error it won't move. Integral gain can be used to generate enough output to the drive to correct the error, but this approach may not be responsive enough for the application. An alternative is to use Backlash Compensation to break sticktion in the presence of a non-zero position error. This is done by adding, or subtracting, a fixed output level, called Backlash Compensation, to the Servo Output value based on its current sign.
The Backlash Compensation value should be just under the value that would break the sticktion. A larger value causes the axis to dither. Dither is when the axis moves rapidly back and forth centered on the commanded position.
This controller attribute is replicated in the motion module.</td></tr>
<tr><td>Backlash Compensation Window</td><td>AXIS_SERVO
AXIS_SERVO_DRIVE</td><td>REAL</td><td>GSV
SSV</td><td>Position Units
To address the issue of dither when applying Backlash Compensation and hunting from the integral gain, a Backlash Compensation Window is applied around the current command position when the axis is not being commanded to move. If the actual position is within the Backlash Compensation Window, the Backlash Compensation value is applied to the Servo Output but scaled by the ratio of the position error to the Backlash Compensation Window. Within the window, the servo integrators are also disabled. Thus, once the position error reaches or exceeds the value of the Backlash Compensation Window attribute, the full Backlash Compensation value is applied. Of course, should the Backlash Compensation Window be set to zero, this feature is effectively disabled.
<table>
<tr><th>Tag</th><th>Bit</th><th>Tag</th><th>Bit</th></tr>
<tr><td>Guard OK</td><td>0</td><td>Guard Max Accel Monitor in Progress</td><td>14</td></tr>
<tr><td>Guard Config Locked</td><td>1</td><td>Guard Direction Monitor in Progress</td><td>15</td></tr>
<tr><td>Guard Gate Drive Output</td><td>2</td><td>Guard Door Control Lock</td><td>16</td></tr>
<tr><td>Guard Stop Input</td><td>3</td><td>Guard Door Control Output</td><td>17</td></tr>
<tr><td>Guard Stop Request</td><td>4</td><td>Guard Door Monitor Input</td><td>18</td></tr>
<tr><td>Guard Stop in Progress</td><td>5</td><td>Guard Door Monitor In Progress</td><td>19</td></tr>
<tr><td>Guard Stop Decel</td><td>6</td><td>Guard Lock Monitor Input</td><td>20</td></tr>
<tr><td>Guard Stop Standstill</td><td>7</td><td>Guard Enabling Switch Input</td><td>21</td></tr>
<tr><td>Guard Stop Output</td><td>8</td><td>Guard Enabling Switch in Progress</td><td>22</td></tr>
<tr><td>Guard Limited Speed Input</td><td>9</td><td>Guard Reset Input</td><td>23</td></tr>
<tr><td>Guard Limited Speed Request</td><td>10</td><td>Guard Reset Required</td><td>24</td></tr>
<tr><td>Guard Limited Speed Monitor in Progress</td><td>11</td><td>Guard Stop Input Cycle Required</td><td>25</td></tr>
<tr><td>Guard Limited Speed Output</td><td>12</td><td>Reserved</td><td>26…31</td></tr>
<tr><td>Guard Max Speed Monitor in Progress</td><td>13</td><td></td><td></td></tr>
</table>
A non-zero Backlash Compensation Window softens the Backlash Compensation as its applied to the Servo Output and reduces the dithering effect that it can create. This generally allows higher values of Backlash Compensation to be applied. Hunting is also eliminated at the cost of a small steady-state error.</td></tr>
</table>

<table>
<tr><th>Attribute</th><th>Axis Type</th><th>Data Type</th><th>Access</th><th>Description</th></tr>
<tr><td>Guard Status</td><td>AXIS_SERVO_DRIVE</td><td>DINT</td><td>Tag
GSV</td><td>Important: To use this attribute, choose it as one of the attributes for Real Time Axis Information for the axis. Otherwise, you won't see the right value as the axis runs. See Axis Info Select 1.
<table>
<tr><th>Tag</th><th>Bit</th><th>Tag</th><th>Bit</th></tr>
<tr><td>Guard OK</td><td>0</td><td>Guard Max Accel Monitor in Progress</td><td>14</td></tr>
<tr><td>Guard Config Locked</td><td>1</td><td>Guard Direction Monitor in Progress</td><td>15</td></tr>
<tr><td>Guard Gate Drive Output</td><td>2</td><td>Guard Door Control Lock</td><td>16</td></tr>
<tr><td>Guard Stop Input</td><td>3</td><td>Guard Door Control Output</td><td>17</td></tr>
<tr><td>Guard Stop Request</td><td>4</td><td>Guard Door Monitor Input</td><td>18</td></tr>
<tr><td>Guard Stop in Progress</td><td>5</td><td>Guard Door Monitor In Progress</td><td>19</td></tr>
<tr><td>Guard Stop Decel</td><td>6</td><td>Guard Lock Monitor Input</td><td>20</td></tr>
<tr><td>Guard Stop Standstill</td><td>7</td><td>Guard Enabling Switch Input</td><td>21</td></tr>
<tr><td>Guard Stop Output</td><td>8</td><td>Guard Enabling Switch in Progress</td><td>22</td></tr>
<tr><td>Guard Limited Speed Input</td><td>9</td><td>Guard Reset Input</td><td>23</td></tr>
<tr><td>Guard Limited Speed Request</td><td>10</td><td>Guard Reset Required</td><td>24</td></tr>
<tr><td>Guard Limited Speed Monitor in Progress</td><td>11</td><td>Guard Stop Input Cycle Required</td><td>25</td></tr>
<tr><td>Guard Limited Speed Output</td><td>12</td><td>Reserved</td><td>26…
31</td></tr>
<tr><td>Guard Max Speed Monitor in Progress</td><td>13</td><td></td><td></td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Attribute</th><th>Axis Type</th><th>Data Type</th><th>Access</th><th>Description</th></tr>
<tr><td>Guard Faults</td><td>AXIS_SERVO_DRIVE</td><td>DINT</td><td>Tag
GSV</td><td>Important: To use this attribute, choose it as one of the attributes for Real Time Axis Information for the axis. Otherwise, you won't see the right value as the axis runs. See Axis Info Select 1.
<table>
<tr><th>Tag</th><th>Bit</th><th>Tag</th><th>Bit</th></tr>
<tr><td>Reserved</td><td>0</td><td>Guard Limited Speed Output Fault</td><td>15</td></tr>
<tr><td>Guard Internal Fault</td><td>1</td><td>Guard Limited Speed Monitor Fault</td><td>16</td></tr>
<tr><td>Guard Configuration Fault</td><td>2</td><td>Guard Max Speed Monitor Fault</td><td>17</td></tr>
<tr><td>Guard Gate Drive Fault</td><td>3</td><td>Guard Max Accel Monitor Fault</td><td>18</td></tr>
<tr><td>Guard Reset Fault</td><td>4</td><td>Guard Direction Monitor Fault</td><td>19</td></tr>
<tr><td>Guard Feedback 1 Fault</td><td>5</td><td>Guard Door Monitor Input Fault</td><td>20</td></tr>
<tr><td>Guard Feedback 2 Fault</td><td>6</td><td>Guard Door Monitor Fault</td><td>21</td></tr>
<tr><td>Guard Feedback Speed Compare Fault</td><td>7</td><td>Guard Door Control Output Fault</td><td>22</td></tr>
<tr><td>Guard Feedback Position Compare Fault</td><td>8</td><td>Guard Lock Monitor Input Fault</td><td>23</td></tr>
<tr><td>Guard Stop Input Fault</td><td>9</td><td>Guard Lock Monitor Fault</td><td>24</td></tr>
<tr><td>Guard Stop Output Fault</td><td>10</td><td>Guard Enabling Switch Monitor Input Fault</td><td>25</td></tr>
<tr><td>Guard Stop Decel Fault</td><td>11</td><td>Guard Enabling Switch Monitor Fault</td><td>26</td></tr>
<tr><td>Guard Stop Standstill Fault</td><td>12</td><td>Guard Feedback 1 Voltage Monitor Fault</td><td>27</td></tr>
<tr><td>Guard Stop Motion Fault</td><td>13</td><td>Guard Feedback 2 Voltage Monitor Fault</td><td>28</td></tr>
<tr><td>Guard Limited Speed Input Fault</td><td>14</td><td>Reserved</td><td>29..
31</td></tr>
</table></td></tr>
<tr><td>Gearing Lock Status</td><td>AXIS_CONSUMED
AXIS_GENERIC
AXIS_SERVO
AXIS_SERVO_DRIVE
AXIS_VIRTUAL</td><td>BOOL</td><td>Tag</td><td>Set whenever the slave axis is locked to the master axis in a gearing relationship according to the specified gear ratio. The clutch function of the gearing planner is used to ramp an axis up, or down, to speed in a gearing process (MAG with Clutch selected). This bit is cleared during the intervals where the axis is clutching.</td></tr>
<tr><td>Gearing Status</td><td>AXIS_CONSUMED
AXIS_GENERIC
AXIS_SERVO
AXIS_SERVO_DRIVE
AXIS_VIRTUAL</td><td>BOOL</td><td>Tag</td><td>Set if the axis is a slave that is currently gearing to another axis. Cleared when the gearing operation is stopped or is superseded by some other motion operation.</td></tr>
<tr><td>Ground Short Fault</td><td>AXIS_SERVO_DRIVE</td><td>BOOL</td><td>Tag</td><td>When the drive detects an imbalance in the DC bus supply current, the Ground Short Fault bit is set, indicating that current is flowing through an improper ground connection.</td></tr>
</table>

Attribute	Axis Type	Data Type	Access	Description
Group Instance	AXIS_CONSUMED AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE AXIS_VIRTUAL	DINT	GSV	Instance Number of Group assigned to Axis The Group Instance attribute is used to determine what motion group object instance this axis is assigned to.
Hard Overtravel Fault Action	AXIS_SERVO_DRIVE	SINT	GSV SSV	**Fault Action** / **Value** Shutdown / 0 Disable Drive / 1 Stop Motion / 2 Status Only / 3
Home Configuration Bits	AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE AXIS_VIRTUAL	DINT	GSV SSV	0 = (Reserved) 1 = Home Switch Normally Closed 2 = Marker Edge Negative **Home Switch Normally Closed** The Home Switch Normally Closed bit attribute determines the normal state of the home limit switch used by the homing sequence. The normal state of the switch is its state before being engaged by the axis during the homing sequence. For example, if the Home Switch Normally Closed bit is set (true) then the condition of the switch before homing is closed. When the switch is engaged by the axis during the homing sequence, the switch is opened, which constitutes a homing event.
Home Direction	AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE AXIS_VIRTUAL	SINT	GSV SSV	0 = unidirectional forward 1 = bidirectional forward 2 = unidirectional reverse 3 = bidirectional reverse
Home Event Armed Status	AXIS_CONSUMED AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE AXIS_VIRTUAL	BOOL	Tag	Set when a home event is armed through execution of the MAH (Motion Axis Home) instruction. Cleared when a home event occurs.
Home Event Status	AXIS_CONSUMED AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE AXIS_VIRTUAL	BOOL	Tag	Set when a home event occurs. Cleared when another MAH (Motion Axis Home) instruction is executed.
Home Event Task	AXIS_CONSUMED AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE AXIS_VIRTUAL	DINT	MSG	User Event Task that is triggered to execute when a Home event occurs. An instance value of 0 indicates that no event task is configured to be triggered by the Home Event. This attribute indicates which user Task is triggered when a home event occurs. The triggering of the user Task occurs simultaneously with the setting of the Process Complete bit for the instruction that armed the home event. This attribute is set through internal communication from the user Task object to the Axis object when the Task trigger attribute is set to select the Home Event Task Instance attribute of the Axis. This attribute should not be set directly by an external device. This attribute is available to be read externally (Get attributes List) for diagnostic information.

Attribute	Axis Type	Data Type	Access	Description
Home Input Status	AXIS_SERVO AXIS_SERVO_DRIVE	BOOL	Tag	If this bit is: • ON — The home input is active. • OFF — The home input is inactive.
Home Mode	AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE AXIS_VIRTUAL	SINT	GSV SSV	0 = passive 1 = active (default) 2 = absolute
Home Offset	AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE AXIS_VIRTUAL	REAL	GSV SSV	Position Units When applied to an active or passive Homing Mode, using a non-immediate Home Sequence, the Home Offset is the desired position offset of the axis Home Position from the position at which the home event occurred. The Home Offset is applied at the end of the specified homing sequence before the axis moves to the Home Position. In most cases, Home Offset is set to zero. After an active bidirectional homing sequence completes, the axis is left at the specified Home Position. If the Home Offset is non-zero, the axis is then offset from the marker or home switch event point by the Home Offset value. If the Home Offset is zero, the axis sits right 'on top of' the marker or home switch point.
Home Position	AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE AXIS_VIRTUAL	REAL	GSV SSV	Position Units The Home Position is the desired absolute position for the axis after the specified homing sequence completes. After an active homing sequence completes, the axis is left at the specified Home Position. In most cases, Home Position is set to zero, although any value, within the Maximum Positive and Negative Travel limits of the axis (if enabled), may also be used. (A description of the Maximum Positive and Negative Travel configuration attributes may be found in the Servo and Drive Axis Object specifications). For a rotary axis, the Home Position is constrained to be a positive number less than the Position Unwind value divided by the Conversion Constant. When configured for absolute Homing Mode, the Home Position value is applied directly to the absolute feedback device to establish an absolute position reference for the system.
Home Return Speed	AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE	REAL	GSV SSV	Position Units / Seconds The Home Return Speed attribute controls the speed of the jog profile used after the first leg of an active bidirectional homing sequence.
Home Sequence	AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE AXIS_VIRTUAL	SINT	GSV SSV	0 = immediate (default) 1 = switch 2 = marker 3 = switch then marker 4 = torque limit 5 = torque limit then marker
Home Speed	AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE	REAL	GSV SSV	Position Units / Seconds The Home Speed attribute controls the speed of the jog profile used in the first leg of an active homing sequence as described in the above discussion of the Home Sequence Type attribute.

Attribute	Axis Type	Data Type	Access	Description
Axis Homed Status	AXIS_CONSUMED AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE AXIS_VIRTUAL	BOOL	Tag	The HomedStatus bit is set by the MAH instruction upon successful completion of the configured homing sequence. This bit indicates that an absolute machine reference position is established. When this bit is set, operations that require a machine reference, such as Software Overtravel checking can be meaningfully enabled. For CIP Drive axis data types, the HomedStatus bit, is cleared under the following conditions: • MRP instruction For non-CIP Drive axis data types, the HomedStatus bit is cleared under the following conditions: • Download • Control power cycle • Reconnection to Motion Module • Feedback Loss Fault • Shutdown
Homing Status	AXIS_CONSUMED AXIS_SERVO AXIS_SERVO_DRIVE AXIS_VIRTUAL	BOOL	Tag	Set if a Home motion profile is currently in progress. Cleared when the homing operation is stopped or is superseded by some other motion operation.
Inhibit Status	AXIS_SERVO AXIS_SERVO_DRIVE	BOOL	Tag	Use the InhibitStatus bit of an axis to see if the axis is inhibited or uninhibited. If the bit is: • ON — The axis is inhibited. • OFF — The axis is uninhibited. The controller changes the InhibitStatus bit only after the following occurs. • The axis changes to inhibited or uninhibited. • All uninhibited axes are ready. • The connections to the motion module are running again.
InhibitAxis	AXIS_SERVO AXIS_SERVO_DRIVE	INT	GSV SSV	**To** / **Set the attribute to** Block the controller from using the axis. This inhibits the axis. / 1 or any non-zero value Let the controller use the axis. This uninhibits the axis. / 0
Integrator Hold Enable	AXIS_SERVO AXIS_SERVO_DRIVE	SINT	GSV SSV	When the Integrator Hold Enable attribute value is configured TRUE, the servo loop temporarily disables any enabled integrators while the command position is changing. This feature is used by point-to-point moves to minimize the integrator wind-up during motion. When the Integrator Hold Enable attribute value is FALSE, all active integrators are enabled. 0 = disabled 1 = enabled
Inter Module Sync Fault	AXIS_SERVO	BOOL	Tag	If this bit is on, the analog servo cards of a SoftLogix5800 controller aren't synchronized. The hardware or vbfirmware of the card causes this fault. For example, the cable between 2 cards isn't connected.
Interpolated Actual Position	AXIS_CONSUMED AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE AXIS_VIRTUAL	REAL	GSV Tag	Interpolated Actual Position in Position Units Interpolated Actual Position is the interpolation of the actual position, based on past axis trajectory history, at the time specified by the 'Interpolated Time' attribute.

Attribute	Axis Type	Data Type	Access	Description
Interpolated Command Position	AXIS_CONSUMED AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE AXIS_VIRTUAL	REAL	GSV Tag	Interpolated Command Position in Position Units Interpolated Command Position is the interpolation of the commanded position, based on past axis trajectory history, at the time specified by the 'Interpolated Time' attribute.
Interpolation Time	AXIS_CONSUMED AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE AXIS_VIRTUAL	DINT	GSV/SSV Tag	CST time to interpolate to Interpolated Time is the 32-bit CST time used to calculate the interpolated positions. When this attribute is updated with a CST value, the Interpolated Actual Position and Interpolated Command Position values are automatically calculated.
Jog Status	AXIS_CONSUMED AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE AXIS_VIRTUAL	BOOL	Tag	Set if a Jog motion profile is currently in progress. Cleared when the Jog is complete or is superseded by some other motion operation.
LDT Calibration Constant	AXIS_SERVO	REAL	GSV	This attribute provides for setting a calibration constant for LDT devices. This attribute is only active if the Transducer Type is set to LDT.
LDT Calibration Constant Units	AXIS_SERVO	SINT	GSV	0 = m/sec 1 = Usec/in This attribute provides a selection for the units of the LDT calibration constant attribute. This attribute is only active if the Transducer Type is set to LDT.
LDT Length	AXIS_SERVO	REAL	GSV	This attribute provides for setting the length of an LDT device. This attribute is only active if the Transducer Type is set to LDT.
LDT Length Units	AXIS_SERVO	SINT	GSV	0 = m 1 = in This attribute provides a selection for the units of the LDT length attribute. This attribute is only active if the Transducer Type is set to LDT.
LDT Recirculations	AXIS_SERVO	SINT	GSV	This attribute provides the number of recirculations. This attribute is only active if the Transducer Type is set to LDT and LDT Type is set to PWM.
LDT Scaling	AXIS_SERVO	REAL	GSV	This attribute provides for setting the scaling factor for LDT devices. This attribute is only active if the Transducer Type is set to LDT.
LDT Scaling Units	AXIS_SERVO	SINT	GSV	0 = Position Units/m 1 = Position Units/in This attribute provides a selection for the units of the LDT scaling attribute. This attribute is only active if the Transducer Type is set to LDT.
LDT Type	AXIS_SERVO	SINT	GSV	0 = PWM 1 = Start/Stop Rising 2 = Start/Stop Falling This attribute provides a selection for the LDT Type. It provides the following enumerated values: PWM, Start/Stop Rising, and Start/Stop Falling. This attribute is only active if the Transducer Type is set to LDT.
Linear Motor Mass	AXIS_SERVO_DRIVE	REAL	SSV	The Linear Motor Mass attribute is a float that specifies the unloaded moving mass of a linear motor.

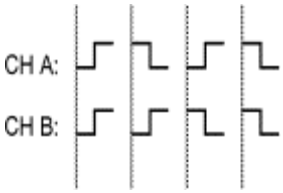
Attribute	Axis Type	Data Type	Access	Description
Linear Motor Rated Speed	AXIS_SERVO_DRIVE	REAL	GSV	The Linear Motor Rated Speed attribute is a float that specifies the nameplate rated speed of a linear motor. For PM motors, this is generally specified at rated voltage based on rated current, rated force, or rated power. For induction motors, this value is the speed of the motor driven at rated frequency under rated force load. This value is synonymous with the term base speed.
Load Inertia Ratio	AXIS_SERVO_DRIVE	REAL	GSV SSV	%Rated / Pos Units per Sec^2 The Motor Inertia value represents the inertia of the motor without any load attached to the motor shaft in Torque Scaling units of %Rated / Pos Units per Sec^2. The Load Inertia Ratio attribute's value represents the ratio of the load inertia to the motor inertia. Auto-tuning uses the Motor Inertia value to calculate the Load Inertia Ratio based on the following equation. Load Inertia Ratio = (Total Inertia - Motor Inertia) / Motor Inertia. Total Inertia is directly measured by the auto-tuning algorithm and applied to the Torque Scaling attribute in units of %Rated / Pos Units per Sec^2. If the Load Inertia Ratio value is known, the Motor Inertia value can also be used to calculate a suitable Torque Scaling value for the fully loaded motor without performing an auto-tune. The equation used by RSLogix5000 to calculate the Torque Scaling value is as follows: Torque Scaling = (1 + Load Inertia Ratio) * Motor Inertia. The value for Load Inertia may be automatically calculated using Rockwell's MotionBook program while the value for Motor Inertia is derived from the Motion database file based on the motor selection.
Map Instance	AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE	DINT	GSV	I/O Map Instance Number. This is 0 for virtual and consumed Data Types. The axis is associated to a motion compatible module by specifying the instance of the map entry representing the module.
Marker Distance	AXIS_SERVO AXIS_SERVO_DRIVE	REAL	GSV Tag	**Important:** To use this attribute, choose it as one of the attributes for Real Time Axis Information for the axis. Otherwise, you won't see the right value as the axis runs. See Axis Info Select 1. Marker Distance in Position Units Marker Distance is the distance between the axis position at which a home switch input was detected and the axis position at which the marker event was detected. This value is useful in aligning a home limit switch relative to a feedback marker pulse to provide repeatable homing operation.

Master Input Configuration Bits	AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE AXIS_VIRTUAL	DINT	GSV SSV	Bits 0 = Master Delay Compensation 1 = Master Position Filter **Master Delay Compensation** By default, the Position Camming and Gearing functions, when applied to a slave axis, perform Master Delay Compensation to compensate for the delay time between reading the master axis command position and applying the associated slave command position to the input of the slave's servo loop. When the master axis is running at a fixed speed, this compensation technique insures that the slave axis command position accurately tracks the actual position of the master axis; in other words, Master Delay Compensation allows for zero tracking error when gearing or camming to the actual position of a master axis. The Master Delay Compensation algorithm extrapolates the position of the master axis at the predicted time when the command position is applied to the slave's servo loop. Because master axis position is measured in discrete feedback counts and is inherently noisy, the extrapolation process amplifies that noise according to the total position update delay. The total position update delay is proportional to the Base Update Period of the motion group, and, if the master or the slave involves an AXIS_SERVO_DRIVE data type, it also includes the delay term that is proportional to the SERCOS Update Period. The greater the delay, the greater the noise introduced by the extrapolator. The Master Delay Compensation feature also has an extrapolation filter to filter the noise introduced by the extrapolation process. The time constant of the filter is fixed at 4x the total position update delay (independent of the Master Position Filter Bandwidth), which again is a function of the Base Update Period (and the SERCOS Update Period, if a AXIS_SERVO_DRIVE data type). The controller uses a 1st order extrapolation algorithm that results in zero tracking error while the master axis is moving at constant velocity. If the master axis accelerates or decelerates the tracking error is non-zero and proportional to the acceleration or deceleration rate and also proportional to the square of the total position update delay time. From a noise and acceleration error perspective, minimizing the base update period is vital. Some applications do not need zero tracking error between the master and the slave axis. In these cases, it may be beneficial to disable the Master Delay Compensation feature to eliminate the disturbances the extrapolation algorithm introduces to the slave axis. When the Master Delay Compensation feature is disabled (bit cleared), the slave axis appears to be more responsive to movements of the master and run generally smoother than when the Master Delay Compensation feature is enabled (bit set). However, when the master axis is running at a constant velocity, the slave lags the master by a tracking error that is proportional to the speed of the master. This function does not get applied when the Master is a **Virtual Axis**. Master Delay Compensation, even if explicitly enabled, is not applied in cases where a slave axis is gearing or camming to the master axis' command position. Because the controller generates the command position directly, there is no intrinsic master position delay to compensate for. **Master Position Filter** The Master Position Filter bit controls the activity of an independent single-pole low-pass filter that effectively filters the specified master axis position input to the slave's gearing or position camming operation. When enabled (bit set), this filter has the effect of smoothing out the actual position signal from the master axis, and thus smoothing out the corresponding motion of the slave axis. The trade-off for smoothness is an increase in lag time between the response of the slave axis to changes in motion of the master. Note that the Master Position Filter also provides filtering to the extrapolation noise introduced by the Master Delay Compensation algorithm, if enabled. When the Master Position Filter bit is set, the bandwidth of the Master Position Filter is controlled by the Master Position Filter Bandwidth attribute, see below. This can be done by setting the Master Position Filter bit and controlling the Master Position Filter Bandwidth directly. Setting the Master Position Filter Bandwidth to zero can be used to effectively disable the filter.

Attribute	Axis Type	Data Type	Access	Description
Master Offset	AXIS_CONSUMED AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE AXIS_VIRTUAL	REAL	GSV Tag	**Important:** To use this attribute, make sure Auto Tag Update is Enabled for the motion group (default setting). Otherwise, you won't see the right value as the axis runs. Master Offset in Master Position Units The Master Offset is the position offset that is currently applied to the master side of the position cam. The Master Offset is returned in master position units. The Master Offset shows the same unwind characteristic as the position of a linear axis.
Master Offset Move Status	AXIS_CONSUMED AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE AXIS_VIRTUAL	BOOL	Tag	Set if a Master Offset Move motion profile is currently in progress. This bit is cleared when the Master Offset Move is complete or is superseded by some other motion operation.
Master Position Filter Bandwidth	AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE AXIS_VIRTUAL	REAL	GSV SSV	Hertz The Master Position Filter Bandwidth attribute controls the activity of the single-pole low-pass filter that filters the specified master axis position input to the slave's gearing or position camming operation. When enabled, this filter smooths out the actual position signal from the master axis, and thus smooths out the corresponding motion of the slave axis. The trade-off for smoothness is an increase in lag time between the response of the slave axis to changes in motion of the master. If the Master Position Filter is disabled, the Master Position Filter Bandwidth does not function.
Maximum Acceleration	AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE AXIS_VIRTUAL	REAL	GSV SSV	Position Units / Sec^2 The Maximum Acceleration and Deceleration attribute values are frequently used by motion instructions such as MAJ, MAM, MCD, to determine the acceleration and deceleration rates to apply to the axis. These instructions all have the option of specifying acceleration and deceleration as a percent of the Maximum Acceleration and Maximum Deceleration attributes for the axis. The Maximum Acceleration and Maximum Deceleration values for the axis are automatically set to ~ 85% of the measured Tune Acceleration and Tune Deceleration by the MAAT (Motion Apply Axis Tune) instruction. If set manually, these values should typically be set to ~85% of the maximum acceleration and maximum deceleration rate of the axis. This provides sufficient 'head-room' for the axis to operate at all times within the acceleration and deceleration limits of the drive and motor.
Maximum Deceleration	AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE AXIS_VIRTUAL	REAL	GSV SSV	Position Units / Sec^2 The Maximum Acceleration and Deceleration attribute values are frequently used by motion instructions such as MAJ, MAM, MCD, to determine the acceleration and deceleration rates to apply to the axis. These instructions all have the option of specifying acceleration and deceleration as a percent of the Maximum Acceleration and Maximum Deceleration attributes for the axis. The Maximum Acceleration and Maximum Deceleration values for the axis are automatically set to ~ 85% of the measured Tune Acceleration and Tune Deceleration by the MAAT (Motion Apply Axis Tune) instruction. If set manually, these values should typically be set to ~85% of the maximum acceleration and maximum deceleration rate of the axis. This provides sufficient 'head-room' for the axis to operate at all times within the acceleration and deceleration limits of the drive and motor.

Attribute	Axis Type	Data Type	Access	Description
Maximum Negative Travel	AXIS_SERVO AXIS_SERVO_DRIVE	REAL	GSV SSV	Position Units The Axis Object provides configurable software travel limits via the Maximum Positive and Negative Travel attributes. If the axis is configured for software overtravel limit checking by setting the Soft Overtravel Bit and the axis passes outside these maximum travel limits, a Software Overtravel Fault is issued. When software overtravel checking is enabled, values for the maximum travel in the Maximum Positive and Maximum Negative Travel attributes need to be established with Maximum Positive Travel greater than Maximum Negative Travel. These values are specified in the configured Position Units of the axis. Soft Travel limits are checked if the Soft Travel Limit enable attribute is true. This controller attribute is replicated in the motion module.
Maximum Positive Travel	AXIS_SERVO AXIS_SERVO_DRIVE	REAL	GSV SSV	Position Units The Axis Object provides configurable software travel limits via the Maximum Positive and Negative Travel attributes. If the axis is configured for software overtravel limit checking by setting the Soft Overtravel Bit and the axis passes outside these maximum travel limits, a Software Overtravel Fault is issued. When software overtravel checking is enabled, values for the maximum travel in the Maximum Positive and Maximum Negative Travel attributes need to be established with Maximum Positive Travel greater than Maximum Negative Travel. These values are specified in the configured Position Units of the axis. Soft Travel limits are checked if the Soft Travel Limit enable attribute is true. This controller attribute is replicated in the motion module
Maximum Speed	AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE AXIS_VIRTUAL	REAL	GSV SSV	Position Units / Sec The value of the Maximum Speed attribute is used by various motion instructions (for example, MAJ, MAM, MCD) to determine the steady-state speed of the axis. These instructions all have the option of specifying speed as a percent of the Maximum Speed attribute value for the axis. The Maximum Speed value for the axis is automatically set to the Tuning Speed by the MAAT (Motion Apply Axis Tune) instruction. This value is typically set to ~90% of the maximum speed rating of the motor. This provides sufficient 'head-room' for the axis to operate at all times within the speed limitations of the motor.
Memory Usage	AXIS_CONSUMED AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE AXIS_VIRTUAL	DINT	MSG	Amount of memory consumed for this instance (in bytes)
Memory Use	AXIS_CONSUMED AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE AXIS_VIRTUAL	INT	GSV	Controller memory space where instance exists. 105 (0x69) = I/O space 106 (0x6a) = Data Table space Logix Designer software uses this attribute to create axis instances in I/O memory for axes that are to be produced or consumed. The Memory Use attribute can only be set as part of an axis create service and is used to control which controller memory the object instance is created in.
Module Channel	AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE	SINT	GSV	Zero based channel number of the module. 0xff, indicates unassigned. The axis is associated to a channel on a motion module by specifying the Module Channel attribute.
Module Class Code	AXIS_SERVO AXIS_SERVO_DRIVE	DINT	GSV	ASA Object class code of the motion engine in the module; for example, 0xAF for the M02AE module. The ASA class code of the object in the motion module that is supporting motion; for example, 0xAF is the ASA object ID of the 'Servo Module Axis Object' residing in the 1756-M02AE module.

<table>
<tr><th>Attribute</th><th>Axis Type</th><th>Data Type</th><th>Access</th><th>Description</th></tr>
<tr><td>Module Fault</td><td>AXIS_CONSUMED
AXIS_GENERIC
AXIS_SERVO
AXIS_SERVO_DRIVE
AXIS_VIRTUAL</td><td>BOOL</td><td>Tag</td><td>Set when a serious fault occurs with the motion module associated with the selected axis. Usually a module fault affects all axes associated with the motion module. A module fault generally results in the shutdown of all associated axes. Reconfiguration of the motion module is required to recover from a module fault condition.
Do you want this fault to give the controller a major fault?
• YES — Set the General Fault Type of the motion group = Major Fault.
• NO — You must write code to handle these faults.</td></tr>
<tr><td>Module Fault Bits</td><td>AXIS_CONSUMED
AXIS_SERVO
AXIS_SERVO_DRIVE</td><td>DINT</td><td>GSV*</td><td>Allows access to the module fault bits in one 32-bit word. This attribute is the same as the Module Faults tag.<table><tr><th>Module Fault</th><th>Bit</th></tr><tr><td>Control Sync Fault</td><td>0</td></tr><tr><td>Module Sync Fault</td><td>1</td></tr><tr><td>Timer Event Fault</td><td>2</td></tr><tr><td>Module Hardware Fault</td><td>3</td></tr><tr><td>SERCOS Ring Fault</td><td>4</td></tr><tr><td>Inter Module Sync Fault</td><td>5</td></tr></table>These faults have module scope instead of axis scope.
• These faults show up in all axes that are connected to the motion module.
• The motion planner updates these fault bits every base update period.
Do you want any of these faults to give the controller a major fault?
• YES — Set the General Fault Type of the motion group = Major Fault.
• NO — You must write code to handle these faults.</td></tr>
<tr><td>Module Faults</td><td>AXIS_SERVO
AXIS_SERVO_DRIVE</td><td>DINT</td><td>Tag</td><td>Allows access to the module fault bits in one 32-bit word. This tag is the same as the Module Fault Bits attribute.<table><tr><th>Module Fault</th><th>Bit</th></tr><tr><td>Control Sync Fault</td><td>0</td></tr><tr><td>Module Sync Fault</td><td>1</td></tr><tr><td>Timer Event Fault</td><td>2</td></tr><tr><td>Module Hardware Fault</td><td>3</td></tr><tr><td>SERCOS Ring Fault</td><td>4</td></tr><tr><td>Inter Module Sync Fault</td><td>5</td></tr></table>These faults have module scope instead of axis scope.
• These faults show up in all axes that are connected to the motion module.
• The motion planner updates these fault bits every base update period.
Do you want any of these faults to give the controller a major fault?
• YES — Set the General Fault Type of the motion group = Major Fault.
• NO — You must write code to handle these faults.</td></tr>
<tr><td>Module Hardware Fault</td><td>AXIS_SERVO
AXIS_SERVO_DRIVE</td><td>BOOL</td><td>Tag</td><td>If this bit is set, there is a hardware issue with the motion module that, in general, is going to require replacement of the module.</td></tr>
</table>

<table>
<tr><th>Attribute</th><th>Axis Type</th><th>Data Type</th><th>Access</th><th>Description</th></tr>
<tr><td>Module Sync Fault</td><td>AXIS_SERVO
AXIS_SERVO_DRIVE</td><td>BOOL</td><td>Tag</td><td>If this bit is set, the motion module lost communication with the controller and missed several position updates in a row.
• The motion module can miss up to 4 position updates. After that, the motion module shuts down.
• This bit clears when communication is reestablished.</td></tr>
<tr><td>Mot Feedback Fault</td><td>AXIS_SERVO_DRIVE</td><td>BOOL</td><td>Tag</td><td>Set for the A Quad B feedback device when one of these happens:
• The differential electrical signals for one or more of the feedback channels (for example, A+ and A-, B+ and B-, or Z+ and Z-) are at the same level (high or low). Under normal operation, the differential signals are at opposite levels. The most common cause of this situation is a broken wire between the feedback transducer and the servo module or drive.
• Loss of feedback ‘power’ or feedback ‘common’ electrical connection between the servo module or drive and the feedback device.
The controller latches this fault. Use a Motion Axis Fault Reset (MAFR) or Motion Axis Shutdown Reset (MASR) instruction to clear the fault.</td></tr>
<tr><td>Motor Feedback Noise Fault</td><td>AXIS_SERVO_DRIVE</td><td>BOOL</td><td>Tag</td><td>Set when there is noise on the feedback device’s signal lines.
• For example, simultaneous transitions of the feedback A and B channels of an A Quad B is referred to generally as feedback noise.
• Feedback noise (shown below) is most often caused by loss of quadrature in the feedback device itself or radiated common-mode noise signals being picked up by the feedback device wiring. You can see these on an oscilloscope.
 

• To troubleshoot the loss of channel quadrature, look for:
  • physical misalignment of the feedback transducer components
  • excessive capacitance (or other delays) on the encoder signals
• Proper grounding and shielding usually cures radiated noise problems.
The controller latches this fault. Use a Motion Axis Fault Reset (MAFR) or Motion Axis Shutdown Reset (MASR) instruction to clear the fault.</td></tr>
<tr><td>Motion Status</td><td>AXIS_CONSUMED
AXIS_GENERIC
AXIS_SERVO
AXIS_SERVO_DRIVE
AXIS_VIRTUAL</td><td>DINT</td><td>Tag</td><td>Allows access to all motion status bits in one 32-bit word. This tag is the same as the Motion Status Bits attribute.
<table>
<tr><th>Motion Status</th><th>Bit</th><th>Motion Status</th><th>Bit</th></tr>
<tr><td>Accel Status</td><td>0</td><td>Time Cam Status</td><td>9</td></tr>
<tr><td>Decel Status</td><td>1</td><td>Position Cam Pending Status</td><td>10</td></tr>
<tr><td>Move Status</td><td>2</td><td>Time Cam Pending Status</td><td>11</td></tr>
<tr><td>Jog Status</td><td>3</td><td>Gearing Lock Status</td><td>12</td></tr>
<tr><td>Gearing Status</td><td>4</td><td>Position Cam Lock Status</td><td>13</td></tr>
<tr><td>Homing Status</td><td>5</td><td>Reserved</td><td>14</td></tr>
<tr><td>Stopping Status</td><td>6</td><td>Master Offset Move Status</td><td>15</td></tr>
<tr><td>Axis Homed Status</td><td>7</td><td>Coordinated Motion Status</td><td>16</td></tr>
<tr><td>Position Cam Status</td><td>8</td><td></td><td></td></tr>
</table></td></tr>
</table>

<table>
<tr><th>Attribute</th><th>Axis Type</th><th>Data Type</th><th>Access</th><th>Description</th></tr>
<tr><td>Motion Status Bits</td><td>AXIS_CONSUMED
AXIS_GENERIC
AXIS_SERVO
AXIS_SERVO_DRIVE
AXIS_VIRTUAL</td><td>DINT</td><td>GSV</td><td>Allows access to all motion status bits in one 32-bit word. This attribute is the same as the Motion Status tag.
<table>
<tr><th>Motion Status</th><th>Bit</th><th>Motion Status</th><th>Bit</th></tr>
<tr><td>Accel Status</td><td>0</td><td>Time Cam Status</td><td>9</td></tr>
<tr><td>Decel Status</td><td>1</td><td>Position Cam Pending Status</td><td>10</td></tr>
<tr><td>Move Status</td><td>2</td><td>Time Cam Pending Status</td><td>11</td></tr>
<tr><td>Jog Status</td><td>3</td><td>Gearing Lock Status</td><td>12</td></tr>
<tr><td>Gearing Status</td><td>4</td><td>Position Cam Lock Status</td><td>13</td></tr>
<tr><td>Homing Status</td><td>5</td><td>Reserved</td><td>14</td></tr>
<tr><td>Stopping Status</td><td>6</td><td>Master Offset Move Status</td><td>15</td></tr>
<tr><td>Axis Homed Status</td><td>7</td><td>Coordinated Motion Status</td><td>16</td></tr>
<tr><td>Position Cam Status</td><td>8</td><td></td><td></td></tr>
</table></td></tr>
<tr><td>Motor Capacity</td><td>AXIS_SERVO_DRIVE</td><td>REAL</td><td>GSV
Tag</td><td>Important: To use this attribute, choose it as one of the attributes for Real Time Axis Information for the axis. Otherwise, you won't see the right value as the axis runs. See Axis Info Select 1.
The present utilization of motor capacity as a percent of rated capacity.</td></tr>
<tr><td>Motor Data</td><td>AXIS_SERVO_DRIVE</td><td>Struct {
INT;
SINT
[256]}</td><td>MSG</td><td>Struct {length; data[ ]}
The Motor Data attribute is a structure with a length element and an array of bytes that contains important motor configuration information needed by an A-B SERCOS drive to operate the motor. The length element represents the number of data elements in the data array. The meaning of data within the data array is understood only by the drive. The block of data stored in the Motor Data attribute is derived at configuration time from a Logix Designer software motion database file.</td></tr>
<tr><td>Motor Electrical Angle</td><td>AXIS_SERVO_DRIVE</td><td>REAL</td><td>GSV
Tag</td><td>Important: To use this attribute, choose it as one of the attributes for Real Time Axis Information for the axis. Otherwise, you won't see the right value as the axis runs. See Axis Info Select 1.
Degrees
The present electrical angle of the motor shaft.</td></tr>
</table>

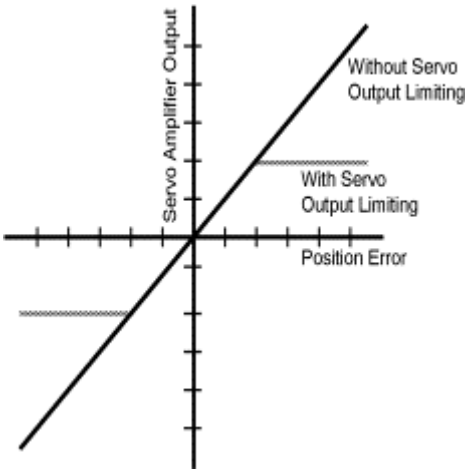
<table>
<tr><th>Attribute</th><th>Axis Type</th><th>Data Type</th><th>Access</th><th>Description</th></tr>
<tr><td>Motor Feedback Configuration</td><td>AXIS_SERVO_DRIVE</td><td>INT</td><td>GSV</td><td>The controller and drive use this for scaling the feedback device counts. These attributes are derived from the corresponding Motor and Auxiliary Feedback Unit attributes.
Bit
0 = Feedback type
0 — rotary (default)
1 — linear
1 = (reserved)
2 = Linear feedback unit
0 — metric
1 — english
3 = Feedback Polarity (Aux Only)
0 — not inverted
1 — inverted
<table><tr><th colspan="3">If the bits are</th><th>Then Feedback Resolution is scaled to</th></tr><tr><th>2</th><th>1</th><th>0</th><th></th></tr><tr><td>0</td><td></td><td>0</td><td>Feedback Cycles per Feedback Rev</td></tr><tr><td>1</td><td></td><td>0</td><td>Feedback Cycles per Feedback Rev</td></tr><tr><td>0</td><td></td><td>1</td><td>Feedback Cycles per mm</td></tr><tr><td>1</td><td></td><td>1</td><td>Feedback Cycles per inch</td></tr></table>Feedback Polarity
The Feedback Polarity bit attribute can be used to change the sense of direction of the feedback device. This bit is only valid for auxiliary feedback devices. When performing motor/feedback hookup diagnostics on an auxiliary feedback device using the MRHD and MAHD instructions, the Feedback Polarity bit is configured for the auxiliary feedback device to insure negative feedback into the servo loop. Motor feedback devices must be wired properly for negative feedback since the Feedback Polarity bit is forced to 0, or non-inverted.</td></tr>
<tr><td>Motor Feedback Interpolation Factor</td><td>AXIS_SERVO_DRIVE</td><td>DINT</td><td>GSV</td><td>Feedback Counts per Cycle
The Feedback Interpolation attributes establish how many Feedback Counts there are in one Feedback Cycle. The Feedback Interpolation Factor depends on the feedback device and the drive feedback circuitry. Quadrature encoder feedback devices and the associated drive feedback interface typically support 4x interpolation, so the Interpolation Factor for these devices would be set to 4 Feedback Counts per Cycle (Cycles are sometimes called Lines). High Resolution Sin/Cosine feedback device types can have interpolation factors as high as 2048 Counts per Cycle. The product to the Feedback Resolution and the corresponding Feedback Interpolation Factor is the overall resolution of the feedback channel in Feedback Counts per Feedback Unit. In our example, a Quadrature encoder with a 2000 line/rev resolution and 4x interpolation factor would have an overall resolution of 8000 counts/rev.</td></tr>
<tr><td>Motor Feedback Resolution</td><td>AXIS_SERVO_DRIVE</td><td>DINT</td><td>GSV</td><td>Cycles per Motor Feedback Unit
The Motor and Aux Feedback Resolution attributes are used to provide the A-B drive with the resolution of the associated feedback device in cycles per feedback unit. These parameters provide the SERCOS drive with critical information needed to compute scaling factors used to convert Drive Counts to Feedback counts.</td></tr>
</table>

<table>
<tr><th>Attribute</th><th>Axis Type</th><th>Data Type</th><th>Access</th><th>Description</th></tr>
<tr><td>Motor Feedback Type</td><td>AXIS_SERVO_DRIVE</td><td>INT</td><td>GSV</td><td>The Motor and Aux Feedback Type attributes are used to identify the motor mounted or auxiliary feedback device connected to the drive.
<table>
<tr><th>Feedback Type</th><th>Code</th><th>Rotary Only</th><th>Linear Only</th><th>Rotary or Linear</th></tr>
<tr><td><None></td><td>0x0000</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>SRS</td><td>0x0001</td><td>X</td><td></td><td></td></tr>
<tr><td>SRM</td><td>0x0002</td><td>X</td><td></td><td></td></tr>
<tr><td>SCS</td><td>0x0003</td><td>X</td><td></td><td></td></tr>
<tr><td>SCM</td><td>0x0004</td><td>X</td><td></td><td></td></tr>
<tr><td>SNS</td><td>0x0005</td><td>X</td><td></td><td></td></tr>
<tr><td>MHG</td><td>0x0006</td><td>X</td><td></td><td></td></tr>
<tr><td>Resolver</td><td>0x0007</td><td>X</td><td></td><td></td></tr>
<tr><td>Analog Reference</td><td>0x0008</td><td>X</td><td></td><td></td></tr>
<tr><td>Sin/Cos</td><td>0x0009</td><td></td><td></td><td>X</td></tr>
<tr><td>TTL</td><td>0x000A</td><td></td><td></td><td>X</td></tr>
<tr><td>UVW</td><td>0x000B</td><td></td><td></td><td>X</td></tr>
<tr><td>Unknown Stegmann</td><td>0x000C</td><td></td><td></td><td>X</td></tr>
<tr><td>Endat</td><td>0x000D</td><td></td><td></td><td>X</td></tr>
<tr><td>RCM21S-4</td><td>0x000E</td><td>X</td><td></td><td></td></tr>
<tr><td>RCM21S-6</td><td>0x000F</td><td>X</td><td></td><td></td></tr>
<tr><td>RCM21S-8</td><td>0x0010</td><td>X</td><td></td><td></td></tr>
<tr><td>LINCODER</td><td>0x0011</td><td></td><td>X</td><td></td></tr>
<tr><td>Sin/Cos with Hall</td><td>0x0012</td><td></td><td></td><td>X</td></tr>
<tr><td>TTL with Hall</td><td>0x0013</td><td></td><td></td><td>X</td></tr>
</table></td></tr>
<tr><td>Motor Feedback Units</td><td>AXIS_SERVO_DRIVE</td><td>INT</td><td>GSV</td><td>The Motor Feedback Units attribute establishes the unit of measure that is applied to the Motor Feedback Resolution attribute value. The Aux Feedback Units attribute establishes the unit of measure that is applied to the Aux Feedback Resolution attribute value. Units appearing in the enumerated list cover linear or rotary, english or metric feedback devices.
0 = revs
1 = inches
2 = mm</td></tr>
<tr><td>Motor ID</td><td>AXIS_SERVO_DRIVE</td><td>INT</td><td>GSV</td><td>The Motor ID attribute contains the enumeration of the A-B motor catalog number associated with the axis. If the Motor ID does not match that of the actual motor, an error is generated during the drive configuration process.</td></tr>
</table>

<table>
<tr><th>Attribute</th><th>Axis Type</th><th>Data Type</th><th>Access</th><th>Description</th></tr>
<tr><td>Motor Inertia</td><td>AXIS_SERVO_DRIVE</td><td>REAL</td><td>GSV
SSV</td><td>%Rated / Pos Units per Sec^2
The Motor Inertia value represents the inertia of the motor without any load attached to the motor shaft in Torque Scaling units of %Rated / Pos Units per Sec^2. The Load Inertia Ratio attribute's value represents the ratio of the load inertia to the motor inertia. Auto-tuning uses the Motor Inertia value to calculate the Load Inertia Ratio based on the following equation.
Load Inertia Ratio = (Total Inertia - Motor Inertia) / Motor Inertia.
Total Inertia is directly measured by the auto-tuning algorithm and applied to the Torque Scaling attribute in units of %Rated / Pos Units per Sec^2.
If the Load Inertia Ratio value is known, the Motor Inertia value can also be used to calculate a suitable Torque Scaling value for the fully loaded motor without performing an auto-tune. The equation used by RSLogix5000 to calculate the Torque Scaling value is as follows:
Torque Scaling = (1 + Load Inertia Ratio) * Motor Inertia.
The value for Load Inertia may be automatically calculated using Rockwell's MotionBook program while the value for Motor Inertia is derived from the Motion database file based on the motor selection.</td></tr>
<tr><td>Motor Overtemp Fault</td><td>AXIS_SERVO_DRIVE</td><td>BOOL</td><td>Tag</td><td>Set when the motor's temperature exceeds the motor shutdown temperature.</td></tr>
<tr><td>Motor Rated Continuous Current</td><td>AXIS_SERVO_DRIVE</td><td>REAL</td><td>GSV</td><td>The Motor Rated Continuous Current attribute is a float that specifies the nameplate AC continuous current rating of the motor. This represents the current applied to the motor under full load conditions at rated speed and voltage. Any positive number. This is a database number and should not be changed.</td></tr>
<tr><td>Motor Rated Peak Current</td><td>AXIS_SERVO_DRIVE</td><td>REAL</td><td>GSV</td><td>The Motor Rated Peak Current attribute is a float that specifies the peak or intermittent current rating of the motor. The peak current rating of the motor is often determined by the thermal constraints of the stator winding or the saturation limits of PM motor magnetic material. Any positive number. This is a database number and should not be changed.</td></tr>
<tr><td>Motor Thermal Fault Action</td><td>AXIS_SERVO_DRIVE</td><td>SINT</td><td>GSV
SSV</td><td><table><tr><th>Fault Action</th><th>Value</th></tr><tr><td>Shutdown</td><td>0</td></tr><tr><td>Disable Drive</td><td>1</td></tr><tr><td>Stop Motion</td><td>2</td></tr><tr><td>Status Only</td><td>3</td></tr></table></td></tr>
<tr><td>Move Status</td><td>AXIS_CONSUMED
AXIS_GENERIC
AXIS_SERVO
AXIS_SERVO_DRIVE
AXIS_VIRTUAL</td><td>BOOL</td><td>Tag</td><td>Set if a Move motion profile is currently in progress. Cleared when the Move is complete or is superseded by some other motion operation.</td></tr>
<tr><td>Neg Dynamic Torque Limit</td><td>AXIS_SERVO_DRIVE</td><td>REAL</td><td>Tag</td><td>The currently operative negative positive torque/current limit magnitude. It should be the lowest value of all torque/current limits in the drive at a given time, including: amplifier peak limit, motor peak limit, user current limit, amplifier thermal limit, and motor thermal limit.</td></tr>
</table>

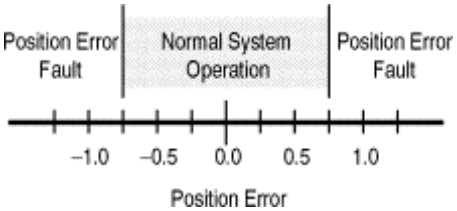
Attribute	Axis Type	Data Type	Access	Description
Neg Hard Overtravel Fault	AXIS_SERVO_DRIVE	BOOL	Tag	Set if the axis moves beyond the negative direction position limits as established by hardware overtravel limit switches mounted on the equipment. This fault can only occur when the drive is in the enabled state and the Hard Overtravel Checking bit is set in the Fault Configuration Bits attribute. If the Hard Overtravel Fault Action is set for Stop Command, the faulted axis can be moved or jogged back inside the soft overtravel limits. Any attempt, however, to move the axis further beyond the hard overtravel limit switch using a motion instruction results in an instruction error. To recover from this fault, the axis must be moved back within normal operation limits of the equipment and the limit switch closed. This fault condition is latched and requires execution of a Motion Axis Fault Reset (MAFR) or Motion Axis Shutdown Reset (MASR) instruction to clear. Any attempt to clear the fault while the overtravel limit switch is still open and the drive is enabled is unsuccessful.
Neg Overtravel Input Status	AXIS_SERVO AXIS_SERVO_DRIVE	BOOL	Tag	If this bit is: • ON — The Negative Overtravel input is active. • OFF — The Negative Overtravel input is inactive.
Neg Soft Overtravel Fault	AXIS_SERVO AXIS_SERVO_DRIVE	BOOL	Tag	If this bit is: • ON — The axis moved or tried to move past the Maximum Negative travel limit. • OFF — The axis moved back within the Maximum Negative travel limit This fault can only happen when the drive is enabled and you configure the axis for Soft Travel Limits. If the Soft Overtravel Fault Action is set for Stop Command, the faulted axis can be moved or jogged back inside the soft overtravel limits. Any attempt, however, to move the axis further beyond the soft overtravel limit using a motion instruction results in an instruction error. As soon as the axis is moved back within the specified soft overtravel limits, the corresponding soft overtravel fault bit is automatically cleared. However, the soft overtravel fault stays through any attempt to clear it while the axis position is still beyond the specified travel limits while the axis is enabled.
Negative Dynamic Torque Limit	AXIS_SERVO_DRIVE	REAL	GSV Tag	**Important:** To use this attribute, choose it as one of the attributes for Real Time Axis Information for the axis. Otherwise, you won't see the right value as the axis runs. See Axis Info Select 1. %Rated The currently operative maximum negative torque/current limit magnitude. The value should be the lowest value of all torque/current limits in the drive at a given time. This limit includes the amplifier peak limit, motor peak limit, user current limit, amplifier thermal limit, and the motor thermal limit.
Output Cam Execution Targets	AXIS_CONSUMED AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE AXIS_VIRTUAL	DINT	GSV	Represents the number of Output Cam nodes attached to this axis. Valid range = 0-8 with default of 0. The Output Cam Execution Targets attribute is used to specify the number of Output Cam nodes attached to the axis. This attribute can only be set as part of an axis create service and dictates how many Output Cam Nodes are created and associated to that axis. Each Output Cam Execution Target requires approximately 5.4k bytes of data table memory to store persistent data. With four Output Cam Execution Targets per axis, an additional 21.6k bytes of memory is required for each axis. The ability to configure the number of Output Cam Execution Targets for an axis reduces the memory required per axis if you do not need Output Cam functionality, or only need 1 or 2 Output Cam Execution Targets for an axis. Each axis can be configured differently.

Attribute	Axis Type	Data Type	Access	Description
Output Cam Lock Status	AXIS_CONSUMED AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE AXIS_VIRTUAL	DINT	GSV Tag	Set of Output Cam Lock Status bits The Output Cam Lock Status bit is set when an Output Cam is armed. This is initiated by executing an MAOC instruction with Immediate execution selected, when a pending output cam changes to armed, or when the axis approaches or passes through the specified axis arm position. As soon as this output cam current position moves beyond the cam start or cam stop position, the Output Cam Lock bit is cleared. This bit is also cleared if the Output Cam is terminated by a MDOC instruction.
Output Cam Lock Status	AXIS_CONSUMED AXIS_SERVO AXIS_SERVO_DRIVE AXIS_VIRTUAL	DINT	Tag	A set of bits that are set when an Output Cam is locked to the Master Axis. The bit number corresponds with the execution target number. One bit per execution target.
Output Cam Pending Status	AXIS_CONSUMED AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE AXIS_VIRTUAL	DINT	GSV Tag	A set of bits that are set when an Output Cam is waiting for an armed Output Cam to move beyond its cam start/cam end position. The bit number corresponds with the execution target number. One bit per execution target. The Output Cam Pending Status bit is set if an Output Cam is currently pending the completion of another Output Cam. This would be initiated by executing an MAOC instruction with Pending execution selected. As soon as this output cam is armed, being triggered when the currently executing Output Cam completes, the Output Cam Pending bit is cleared. This bit is also cleared if the Output Cam is terminated by an MDOC instruction.
Output Cam Status	AXIS_CONSUMED AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE AXIS_VIRTUAL	DINT	GSV Tag	A set of bits that are set when the Output Cam is initiated. The bit number corresponds with the execution target number. One bit per execution target. The Output Cam Status bit is set when an Output Cam is initiated. The Output Cam Status bit is reset when the cam position moves beyond the cam start or cam end position in 'Once' execution mode with no Output Cam pending or when the Output Cam is terminated by an MDOC instruction.
Output Cam Transition Status	AXIS_CONSUMED AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE AXIS_VIRTUAL	DINT	GSV Tag	A set of bits that are set when the transition from the current armed Output Cam to the pending Output Cam is in process. The bit number corresponds with the execution target number. One bit per execution target. The Output Cam Transition Status bit is set when a transition between the currently armed and the pending Output Cam is in process. Therefore, each Output Cam controls a subset of Output Bits. The Output Cam Transition Status bit is reset when the transition to the pending Output Cam is complete or when the Output Cam is terminated by an MDOC instruction.

Attribute	Axis Type	Data Type	Access	Description
Output Limit	AXIS_SERVO	REAL	GSV SSV	0.0…10.0V This controller attribute is replicated in the motion module. The Output Limit attribute provides a method of limiting the maximum servo output voltage of a physical axis to a specified level. The servo output for the axis as a function of position servo error, with and without servo output limiting, is shown below.  The servo output limit may be used as a software current or torque limit if you are using a servo drive in torque (current) loop mode. The percentage of the drive's maximum current that the servo controller commands is equal to the specified servo output limit. For example, if the drive is capable of 30 Amps of current for a 10 Volt input, setting the servo output limit to 5V limits the maximum drive current to 15 Amps. The servo output limit may also be used if the drive cannot accept the full ±10 Volt range of the servo output. In this case, the servo output limit value effectively limits the maximum command sent to the amplifier. For example, if the drive can only accept command signals up to ±7.5 Volts, set the servo output limit value to 7.5 volts.
Output Limit Status	AXIS_SERVO	BOOL	Tag	If this bit is: • ON — The servo output is at or past the Output Limit value. • OFF — The servo output is within the Output Limit value
Output LP Filter Bandwidth	AXIS_SERVO AXIS_SERVO_DRIVE	REAL	GSV SSV	Hertz The Output LP (Low Pass) Filter Bandwidth controls the bandwidth of the drive's low-pass digital output filter. The programmable low-pass output filter is bypassed if the configured Output LP Filter Bandwidth for this filter is set to zero (the default). This output filter can be used to filter out, or reduce, high frequency variation of the drive output to the motor. The lower the Output LP Filter Bandwidth, the greater the attenuation of these high frequency components of the output signal. Unfortunately, since the low-pass filter adds lag to the servo loop, which pushes the system towards instability, decreasing the Output LP Filter Bandwidth usually requires lowering the Position or Velocity Proportional Gain of the system to maintain stability. The output filter is particularly useful in high inertia applications where resonance behavior can severely restrict the maximum bandwidth capability of the servo loop. This controller attribute is replicated in the motion module.

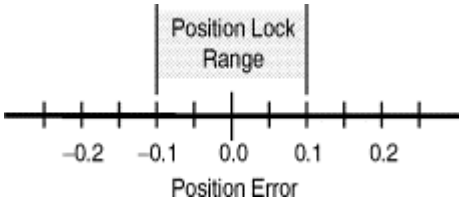
Attribute	Axis Type	Data Type	Access	Description
Output Notch Filter Frequency	AXIS_SERVO_DRIVE	REAL	GSV SSV	Hertz The Output Notch Filter Frequency attribute controls the center frequency of the drive's digital notch filter. Currently implemented as a 2nd order digital filter with a fixed Q, the Notch Filter provides approximately 40DB of output attenuation at the Notch Filter Frequency. The programmable notch filter is bypassed if the configured Output Notch Filter Frequency for this filter is set to zero (the default). This output notch filter is particularly useful in attenuating mechanical resonance phenomena. The output filter is particularly useful in high inertia applications where mechanical resonance behavior can severely restrict the maximum bandwidth capability of the servo loop.
Output Offset	AXIS_SERVO	REAL	GSV SSV	+/-10V Another common situation when interfacing an external Servo Drive, particularly for velocity servo drives, is the effect of drive offset. Cumulative offsets of the servo module's DAC output and the Servo Drive Input result in a situation where a zero commanded Servo Output value causes the axis to 'drift'. If the drift is excessive, it can play havoc on the Hookup Diagnostic and Tuning procedures and result in a steady-state non-zero position error when the servo loop is closed. Output offset compensation can be used to correct this problem by adding a fixed value, called Output Offset, to the Servo Output. This value is chosen to achieve near zero drive velocity when the uncompensated Servo Output value is zero. This controller attribute is replicated in the motion module.
Overload Fault	AXIS_SERVO_DRIVE	BOOL	Tag	When the load limit of the motor/drive is first exceeded, the Overload warning bit is set. If the condition persists, the Overload fault is set. Often this bit is tied into the IT limit of the drive.
Overspeed Fault	AXIS_SERVO_DRIVE	BOOL	Tag	Set when the speed of the axis, as determined from the feedback, exceedes the overspeed limit, which is typically set to 150% of configured velocity limit for the motor.
Physical Axis Fault	AXIS_CONSUMED AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE AXIS_VIRTUAL	BOOL	Tag	If this bit is set, there are one or more fauilts on the physical axis. The faults can then be determined through access to the fault attributes of the associated physical axis. Do you want this fault to give the controller a major fault? • YES — Set the General Fault Type of the motion group = Major Fault. • NO — You must write code to handle these faults.
Pos Dynamic Torque Limit	AXIS_SERVO_DRIVE	REAL	Tag	The currently operative maximum positive torque/current limit magnitude. It should be the lowest value of all torque/current limits in the drive at a given time, including: amplifier peak limit, motor peak limit, user current limit, amplifier thermal limit, and motor thermal limit.
Pos Hard Overtravel Fault	AXIS_SERVO_DRIVE	BOOL	Tag	Set if the axis moves beyond the current position limits as established by hardware overtravel limit switches mounted on the equipment. This fault can only occur when the drive is in the enabled state and the Hard Overtravel Checking bit is set in the Fault Configuration Bits attribute. If the Hard Overtravel Fault Action is set for Stop Command, the faulted axis can be moved or jogged back inside the soft overtravel limits. Any attempt, however, to move the axis further beyond the hard overtravel limit switch using a motion instruction results in an instruction error. To recover from this fault, the axis must be moved back within normal operation limits of the equipment and the limit switch closed. This fault condition is latched and requires execution of an Motion Axis Fault Reset (MAFR) or Motion Axis Shutdown Reset (MASR) instruction to clear. Any attempt to clear the fault while the overtravel limit switch is still open and the drive is enabled is unsuccessful.
Pos Lock Status	AXIS_SERVO AXIS_SERVO_DRIVE	DINT	Tag	Set when the magnitude of the axis position error is less than or equal to the configured Position Lock Tolerance value for the associated physical axis.

Attribute	Axis Type	Data Type	Access	Description
Pos Overtravel Input Status	AXIS_SERVO AXIS_SERVO_DRIVE	BOOL	Tag	If this bit is: • ON — The Positive Overtravel input is active. • OFF — The Positive Overtravel input is inactive.
Pos Soft Overtravel Fault	AXIS_SERVO AXIS_SERVO_DRIVE	BOOL	Tag	If this bit is: • ON — The axis moved or tried to move past the Maximum Positive travel limit. • OFF — The axis moved back within the Maximum Positive travel limit This fault can only happen when the drive is enabled and you configure the axis for Soft Travel Limits. If the Soft Overtravel Fault Action is set for Stop Command, the faulted axis can be moved or jogged back inside the soft overtravel limits. Any attempt, however, to move the axis further beyond the soft overtravel limit using a motion instruction results in an instruction error. As soon as the axis is moved back within the specified soft overtravel limits, the corresponding soft overtravel fault bit is automatically cleared. However, the soft overtravel fault stays through any attempt to clear it while the axis position is still beyond the specified travel limits while the axis is enabled.
Position Cam Lock Status	AXIS_CONSUMED AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE AXIS_VIRTUAL	BOOL	Tag	Set whenever the master axis satisfies the starting condition of a currently active Position Cam motion profile. The starting condition is established by the Start Control and Start Position parameters of the MAPC instruction. This bit is cleared when the current position cam profile completes, or is superseded by some other motion operation. In unidirectional master direction mode, the Position Cam Lock Status bit is cleared when moving in the 'wrong' direction and sets when moving in the 'correct' direction.
Position Cam Pending Status	AXIS_CONSUMED AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE AXIS_VIRTUAL	BOOL	Tag	Set if a Position Cam motion profile is currently pending the completion of a currently executing cam profile. This would be initiated by executing an MAPC instruction with Pending execution selected. This bit is cleared when the current position cam profile completes, initiating the start of the pending cam profile. This bit is also cleared if the position cam profile completes, or is superseded by some other motion operation.
Position Cam Status	AXIS_CONSUMED AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE AXIS_VIRTUAL	BOOL	Tag	Set if a Position Cam motion profile is currently in progress. Cleared when the Position Cam is complete or is superseded by some other motion operation.
Position Command	AXIS_SERVO AXIS_SERVO_DRIVE	REAL	GSV Tag	Position Command in Position Units **Important:** To use this attribute, choose it as one of the attributes for Real Time Axis Information for the axis. Otherwise, you won't see the right value as the axis runs. See Axis Info Select 1. Position Command is the current value of the Fine Command Position into the position loop summing junction, in configured axis Position Units. Within the active servo loop, the Position Command value is used to control the position of the axis.
Position Data Scaling	AXIS_SERVO_DRIVE	INT	GSV	This attribute is derived from the Drive Units attribute. See IDN 76 in IEC 1491.
Position Data Scaling Exp	AXIS_SERVO_DRIVE	INT	GSV	This attribute is derived from the Drive Units attribute. See IDN 78 in IEC 1491.
Position Data Scaling Factor	AXIS_SERVO_DRIVE	DINT	GSV	This attribute is derived from the Drive Units attribute. See IDN 77 in IEC 1491.

<table>
<tr><th>Attribute</th><th>Axis Type</th><th>Data Type</th><th>Access</th><th>Description</th></tr>
<tr><td>Position Differential Gain</td><td>AXIS_SERVO</td><td>REAL</td><td>GSV
SSV</td><td>In some External Velocity Servo Drive applications where the level of damping provided by the external drive is insufficient for good position servo loop performance, additional damping may be achieved via the Position Loop Differential Gain. Assuming a non-zero Position Loop Differential Gain value, the difference between the current Position Error value and the last Position Error value is computed. This value is then multiplied by the Position Loop Differential Gain to produce a component to the Servo Output or Velocity Command that attempts to correct for the change in position error, creating a ‘damping’ effect. Increasing this gain value results in greater ‘damping’ of the axis.</td></tr>
<tr><td>Position Error</td><td>AXIS_SERVO
AXIS_SERVO_DRIVE</td><td>REAL</td><td>GSV
Tag</td><td>Important: To use this attribute, choose it as one of the attributes for Real Time Axis Information for the axis. Otherwise, you won’t see the right value as the axis runs. See Axis Info Select 1.
Position Error in Position Units
Position Error is the difference, in configured axis Position Units, between the command and actual positions of an axis. For an axis with an active servo loop, position error is used, along with other error terms, to drive the motor to the condition where the actual position is equal to the command position.</td></tr>
<tr><td>Position Error Fault</td><td>AXIS_SERVO
AXIS_SERVO_DRIVE</td><td>BOOL</td><td>Tag</td><td>Set when the axis position error exceeds the Position Error Tolerance. This fault can only occur when the drive is in the enabled state.
The controller latches this fault. Use a Motion Axis Fault Reset (MAFR) or Motion Axis Shutdown Reset (MASR) instruction to clear the fault.</td></tr>
<tr><td>Position Error Fault Action</td><td>AXIS_SERVO
AXIS_SERVO_DRIVE</td><td>SINT</td><td>GSV
SSV</td><td><table><tr><th>Fault Action</th><th>Value</th></tr><tr><td>Shutdown</td><td>0</td></tr><tr><td>Disable Drive</td><td>1</td></tr><tr><td>Stop Motion</td><td>2</td></tr><tr><td>Status Only</td><td>3</td></tr></table>This controller attribute is replicated in the motion module.</td></tr>
<tr><td>Position Error Tolerance</td><td>AXIS_SERVO
AXIS_SERVO_DRIVE</td><td>REAL</td><td>GSV
SSV</td><td>Position Units
The Position Error Tolerance parameter specifies how much position error the servo or drive tolerates before issuing a Position Error Fault. Like the position lock tolerance, the position error tolerance is interpreted as a ± quantity. For example, specifying a position error tolerance of 0.75 Position Units means that a Position Error Fault is generated whenever the position error of the axis is greater than 0.75 or less than -0.75 Position Units, as shown.
 

The self tuning routine sets the position error tolerance to twice the following error at maximum speed based on the measured response of the axis. In most applications, this value provides reasonable protection in case of an axis fault or stall condition without nuisance faults during normal operation. If you need to change the calculated position error tolerance value, the recommended setting is 150% to 200% of the position error while the axis is running at its maximum speed.
This controller attribute is replicated in the motion module.</td></tr>
</table>

Attribute	Axis Type	Data Type	Access	Description
Position Feedback	AXIS_SERVO AXIS_SERVO_DRIVE	REAL	GSV Tag	**Important:** To use this attribute, choose it as one of the attributes for Real Time Axis Information for the axis. Otherwise, you won't see the right value as the axis runs. See Axis Info Select 1. Position Feedback in Position Units Position Feedback is the current value of the Fine Actual Position into the position loop summing junction, in configured axis Position Units. Within the servo loop, the Position Feedback represents the current position of the axis.
Position Integral Gain	AXIS_SERVO AXIS_SERVO_DRIVE	REAL	GSV SSV	1/mSec-Sec This controller attribute is replicated in the motion module. Position Integral Gain (Pos I Gain) improves the steady-state positioning performance of the system. By using Position Integral Gain, it is possible to achieve accurate axis positioning despite the presence of such disturbances as static friction or gravity. Increasing the integral gain generally increases the ultimate positioning accuracy of the system. Excessive integral gain, however, results in system instability. Every servo update, the current Position Error is accumulated in a variable called the Position Integral Error. This value is multiplied by the Position Integral Gain to produce a component to the Velocity Command that attempts to correct for the position error. The characteristic of Pos I Gain correction, however, is that any non-zero Position Error accumulates in time to generate enough force to make the correction. This attribute of Pos I Gain makes it invaluable in applications where positioning accuracy or tracking accuracy is critical. The higher the Pos I Gain value the faster the axis is driven to the zero Position Error condition. Unfortunately, Pos I Gain control is intrinsically unstable. Too much Pos I Gain results in axis oscillation and servo instability. If the axis is configured for an external velocity loop servo drive, the Pos I Gain should be zero–most analog velocity loop servo amplifiers have integral gain of their own and do not tolerate *any* amount of Pos I Gain in the position loop without producing severe oscillations. If Pos I Gain is necessary for the application, the velocity integrator in the drive must be disabled. In certain cases, Pos I Gain control is disabled. One such case is when the servo output to the axis' drive is saturated. Continuing integral control behavior in this case would only exacerbate the situation. Another common case is when performing certain motion. When the Integrator Hold Enable attribute is set, the servo loop automatically disables the integrator during commanded motion. While the Pos I Gain, if employed, is typically established by the automatic servo tuning procedure, the Pos I Gain value may also be set manually. You can compute the Pos I Gain based on the current or computed value for the Pos P Gain using the following formula: Pos I Gain = 0.25 * 0.001 Sec/mSec * (Pos P Gain)^2 Assuming a Pos P Gain value of 100 Sec-1, this results in a Pos I Gain value of 2.5 ~0.1 mSec^{-1}-Sec^{-1}
Position Integrator Error	AXIS_SERVO AXIS_SERVO_DRIVE	REAL	GSV Tag	**Important:** To use this attribute, choose it as one of the attributes for Real Time Axis Information for the axis. Otherwise, you won't see the right value as the axis runs. See Axis Info Select 1. Position Integrator Error in Position Units - mSec Position Integrator Error is the running sum of the Position Error, in the configured axis Position Units, for the specified axis. For an axis with an active servo loop, the position integrator error is used, along with other error terms, to drive the motor to the condition where the actual position is equal to the command position.
Position Lock Status	AXIS_SERVO AXIS_SERVO_DRIVE	BOOL	Tag	If this bit is: • ON — The axis position error is less than or equal to the Position Lock Tolerance value of the axis. • OFF — The axis position error is greater than the Position Lock Tolerance value of the axis.

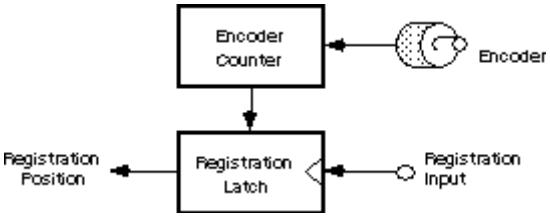
Attribute	Axis Type	Data Type	Access	Description
Position Lock Tolerance	AXIS_SERVO AXIS_SERVO_DRIVE	REAL	GSV SSV	Position Units The Position Lock Tolerance attribute value specifies how much position error the motion module tolerates when giving a true Position Locked Status indication. When used in conjunction with the Position Locked Status bit, it is a useful parameter to control positioning accuracy. The Position Lock Tolerance value should be set, in Position Units, to the desired positioning accuracy of the axis. Note that the position lock tolerance value is interpreted as a ± quantity. For example, if your position units are Inches, specifying a position lock tolerance of 0.01 provides a minimum positioning accuracy of ±0.01 inches as shown below. 
Position Polarity	AXIS_SERVO_DRIVE	INT	GSV	This attribute is derived from the Drive Polarity attribute. See IDN 55 in IEC 1491.

Attribute	Axis Type	Data Type	Access	Description
Position Proportional Gain	AXIS_SERVO AXIS_SERVO_DRIVE	REAL	GSV SSV	1/Seconds This controller attribute is replicated in the motion module. The Position Error is multiplied by the Position Proportional Gain (Pos P Gain) to produce a component to the Velocity Command that tries to correct for the position error. Increasing this gain increases the bandwidth of the position servo loop and results in greater static stiffness of the axis, which is a measure of the corrective force that is applied to an axis for a given position error. Too little Pos P Gain results in excessively compliant, or mushy, axis behavior. Too large a Pos P Gain results in axis oscillation due to servo instability. A well-tuned system moves and stops quickly and shows little or no ringing during constant velocity or when the axis stops. If the response time is poor, or the motion sloppy or slow, you may need to increase the proportional gain. If excessive ringing or overshoot is observed when the motor stops, you may need to decrease the proportional gain. While the tuning procedure sets the Pos P Gain, you can also set it manually. You can compute the Pos P Gain based on the desired loop gain or the desired bandwidth of the position servo system. **Loop Gain Method** If you know the desired loop gain in Inches per Minute per mil or millimeters per minute per mil, use the following formula to calculate the corresponding P gain. Pos P Gain = 16.667 * Desired Loop Gain (IPM/mil) A loop gain of 1 IPM/mil (Pos P gain = 16.7 Sec-1) gives stable positioning for most axes. However, position servo systems typically run much tighter than this. The typical value for the Position Proportional Gain is ~100 Sec-1. **Bandwidth Method** If you know the desired unity gain bandwidth of the position servo in Hertz, use the following formula to calculate the corresponding P gain. Pos P Gain = Bandwidth (Hertz) / 6.28 Position servo systems typically run with at least a unity gain bandwidth of ~16 Hertz. The typical value for the Position Proportional Gain is ~100 Sec-1. **Maximum Bandwidth** There are limitations to the maximum bandwidth that can be achieved for the position loop based on the dynamics of the inner velocity and torque loops of the system and the desired damping of the system, Z. These limitations may be expressed as follows: Bandwidth (Pos) = 0.25 * $1/Z^2$ * Bandwidth (Vel) = 0.25 * $1/Z^2$ * Bandwidth (Torque) For example, if the bandwidth of the drive's torque loop is 100 Hz and the damping factor, Z, is 0.8, the velocity bandwidth is approximately 40 Hz and the position bandwidth is 16 Hz. Based on these numbers, the corresponding proportional gains for the loops can be computed. Note that the bandwidth of the torque loop includes feedback sampling delay and filter time constant.

Attribute	Axis Type	Data Type	Access	Description
Position Servo Bandwidth	AXIS_SERVO AXIS_SERVO_DRIVE	REAL	GSV SSV	Hertz The value for the Position Servo Bandwidth represents the unity gain bandwidth that is to be used to calculate the gains for a subsequent MAAT (Motion Apply Axis Tune) instruction. The unity gain bandwidth is the frequency beyond which the position servo is unable to provide any significant position disturbance correction. In general, within the constraints of a stable servo system, the higher the Position Servo Bandwidth is the better the dynamic performance of the system. A maximum value for the Position Servo Bandwidth is generated by the MRAT (Motion Run Axis Tune) instruction. Computing gains based on this maximum value via the MAAT instruction results in dynamic response in keeping with the current value of the Damping Factor described above. Alternatively, the responsiveness of the system can be 'softened' by reducing the value of the Position Servo Bandwidth before executing the MAAT instruction. There are limitations to the maximum bandwidth that can be achieved for the position loop based on the dynamics of the inner velocity and current loops of the servo system and the desired damping of the system, Z. Exceeding these limits could result in an unstable system. These bandwidth limitations may be expressed as follows: Max Position Bandwidth (Hz) = 0.25 * 1/Z2 * Velocity Bandwidth (Hz) For example, if the maximum bandwidth of the velocity servo loop is 40 Hz and the damping factor, Z, is 0.8, the maximum position bandwidth is 16 Hz. Based on these numbers, the corresponding proportional gains for the loops can be computed.
Position Units	AXIS_CONSUMED AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE AXIS_VIRTUAL	STRING	MSG	Fixed length string of 32 characters The Position Units attribute can support an ASCII text string of up to 32 characters. This string is used by Logix Designer software in the axis configuration dialog boxes to request values for motion-related parameters in the specified Position Units.
Position Unwind	AXIS_CONSUMED AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE AXIS_VIRTUAL	DINT	GSV SSV	Counts per Revolution This controller attribute is replicated in the motion module. If the axis is configured as a rotary axis by setting the corresponding Rotary Axis bit Servo Configuration Bit word, a value for the Position Unwind attribute is required. This is the value used to perform automatic electronic unwind of the rotary axis. Electronic unwind allows infinite position range for rotary axes by subtracting the unwind value from the actual and command position every time the axis makes a complete revolution. To avoid accumulated error due to round-off with irrational conversion constants, the unwind value is requested in units feedback counts per axis revolution and must be an integer. For example, suppose that a given axis is configured as a Rotary Axis with Position Units of 'Degrees' and 10 feedback counts per degree. It is desired to unwind the axis position after every revolution. In this case, the Position Unwind attribute should be set to 3600 since there are 3600 feedback counts (10 * 360) per revolution of the axis.
Positive Dynamic Torque Limit	AXIS_SERVO_DRIVE	REAL	GSV Tag	**Important:** To use this attribute, choose it as one of the attributes for Real Time Axis Information for the axis. Otherwise, you won't see the right value as the axis runs. See Axis Info Select 1. %Rated The currently operative maximum positive torque/current limit magnitude. The value should be the lowest value of all torque/current limits in the drive at a given time. This limit includes the amplifier peak limit, motor peak limit, user current limit, amplifier thermal limit, and the motor thermal limit.
Power Capacity	AXIS_SERVO_DRIVE	REAL	GSV Tag	**Important:** To use this attribute, choose it as one of the attributes for Real Time Axis Information for the axis. Otherwise, you won't see the right value as the axis runs. See Axis Info Select 1. The present utilization of the axis power supply as a percent of rated capacity.

Attribute	Axis Type	Data Type	Access	Description
Power Limit Status	AXIS_SERVO_DRIVE	BOOL	Tag	Set when the magnitude of the actual supplied power is greater than the configured Power Threshold.
Power Phase Loss Fault	AXIS_SERVO_DRIVE	BOOL	Tag	Set when the drive detects that one or more of the three power line phases is lost from the 3 phase power inputs.
Power Supply ID	AXIS_SERVO_DRIVE	INT	GSV	The Power Supply ID attribute contains the enumeration of the A-B Power Supply or System Module catalog numbers associated with the axis. If the Power Supply ID does not match that of the actual supply hardware, an error is generated during the drive configuration process.
Precharge Overload Fault	AXIS_SERVO_DRIVE	BOOL	Tag	The drive's pre-charge resistor gets too hot if you cycle 3-phase power too many times. If that happens, this bit turns on.
Primary Operation Mode	AXIS_SERVO_DRIVE	INT	GSV	This attribute is derived from the Servo Loop Configuration attribute. See IDN 32 in IEC 1491.
Process Status	AXIS_SERVO AXIS_SERVO_DRIVE	BOOL	Tag	Set when there is an axis tuning operation or an axis hookup diagnostic test operation in progress on the axis.
Programmed Stop Mode	AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE AXIS_VIRTUAL	SINT	GSV SSV	Determines how an axis stops when there is a critical controller mode change or when an MGS (Motion Group Stop) instruction executes with its stop mode set to Programmed. The modes of the controller are: Program Mode, Run Mode, Test Mode, and Faulted Mode. Any mode change into or out of program mode (prog->run, prog->test, run->prog & test->prog) initiates a programmed stop for every axis owned by that controller. Each individual axis can have its own Programmed Stop Mode configuration independent of other axes. **Fast Stop (default) = 0** When the Programmed Stop Mode attribute is configured for Fast Stop, the axis is decelerated to a stop using the current configured value for Maximum Deceleration. Servo action is maintained after the axis motion stopped. **Fast Disable = 1** When the Programmed Stop Mode attribute is configured for Fast Disable, the axis is decelerated to a stop using the current configured value for Maximum Deceleration. Servo action is maintained until the axis motion stopped at which time the axis is disabled, that is, Drive Enable disabled, and Servo Action disabled **Hard Disable = 2** When configured for Hard Disable, the axis is immediately disabled, that is, Drive Enable disabled, Servo Action disabled, but the OK contact is left closed. Unless the drive is configured to provide some form of dynamic breaking, this results in the axis coasting to a stop. **Fast Shutdown = 3** When configured for Fast Shutdown, the axis is decelerated to a stop as with Fast Stop. But, once the axis motion is stopped, the axis is placed in the Shutdown state, that is, Drive Enable disabled, servo action disabled, and the OK contact opened. To recover from the Shutdown state requires execution of one of the axis or group Shutdown Reset instructions (MASR or MGSR). **Hard Shutdown = 4** When configured for Hard Shutdown, the axis is immediately placed in the Shutdown state, that is, Drive Enable disabled, Servo Action disabled, and the OK contact opened. Unless the drive is configured to provide some form of dynamic breaking, this results in the axis coasting to a stop. To recover from the Shutdown state requires execution of one of the axis or group Shutdown Reset instructions (MASR or MGSR).

Attribute	Axis Type	Data Type	Access	Description
PWM Frequency Select	AXIS_SERVO_DRIVE	SINT	GSV	The PWM Frequency Select attribute controls the frequency of the pulse width modulated voltage applied to the motor by the drive's power structure. Higher PWM Frequency values reduce torque ripple and motor noise based on the motor's electrical time constant. Higher PWM frequencies, however, mean higher switching frequencies, which tends to produce more heat in the drive's power structure. So, for applications that have high torque demands, a lower PWM frequency is recommended. 0 = low frequency (default) 1 = high frequency
Reg 1 Input Status	AXIS_SERVO AXIS_SERVO_DRIVE	BOOL	Tag	If this bit is: • ON — Registration 1 input is active. • OFF — Registration 1 input is inactive.
Reg 2 Input Status	AXIS_SERVO AXIS_SERVO_DRIVE	BOOL	Tag	If this bit is: • ON — Registration 2 input is active. • OFF — Registration 2 input is inactive.
Reg Event 1 Armed Status	AXIS_CONSUMED AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE AXIS_VIRTUAL	BOOL	Tag	Set when a registration checking is armed for registration input 1 through execution of the MAR (Motion Arm Registration) instruction. Cleared when a registration event occurs or an MDR (Motion Disarm Registration) instruction is executed for registration input 1.
Reg Event 1 Status	AXIS_CONSUMED AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE AXIS_VIRTUAL	BOOL	Tag	Set when a registration event occurs on registration input 1. Cleared when another MAR (Motion Arm Registration) instruction or an MDR (Motion Disarm Registration) instruction is executed for registration input 1.
Reg Event 2 Armed Status	AXIS_CONSUMED AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE AXIS_VIRTUAL	BOOL	Tag	Set when a registration checking is armed for registration input 2 through execution of the MAR (Motion Arm Registration) instruction. Cleared when a registration event occurs or an MDR (Motion Disarm Registration) instruction is executed for registration input 2.
Reg Event 2 Status	AXIS_CONSUMED AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE AXIS_VIRTUAL	BOOL	Tag	Set when a registration event occurs on registration input 2. Cleared when another MAR (Motion Arm Registration) instruction or an MDR (Motion Disarm Registration) instruction is executed for registration input 2.
Registration 1 Position	AXIS_CONSUMED AXIS_SERVO_DRIVE AXIS_VIRTUAL	REAL	Tag	Registration 1 Position in Position Units
Registration 1 Event Task Registration 2 Event Task	AXIS_CONSUMED AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE AXIS_VIRTUAL	DINT	MSG	The following attributes show which task is triggered when the registration event happens. • An instance of 0 means that no event task is configured to be triggered by the registration event. • The triggering of the task occurs simultaneously with the setting of the Process Complete bit for the instruction that armed the watch event. • The controller sets these attributes. Do not set them by an external device.

Attribute	Axis Type	Data Type	Access	Description
Registration 1 Position Registration 2 Position	AXIS_CONSUMED AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE AXIS_VIRTUAL	REAL	GSV Tag	Position Units Two registration position attributes are provided to independently store axis position associated with two registration input events. The Registration Position value is the absolute position of a physical or virtual axis (in the position units of that axis) at the occurrence of the most recent registration event for that axis. The figure below shows how the registration position is latched by the registration input when a registration event occurs. The latching mechanism can be implemented in the controller software (soft registration) or, for greater accuracy, in physical hardware (hard registration).  The Registration Latch mechanism is controlled by two Event Control instructions, MAR (Motion Arm Registration) and MDR (Motion Disarm Registration). The accuracy of the registration position value, saved as a result of a registration event, is a function of the delay in recognizing the specified transition (typically 1 µsec for hardware registration) and the speed of the axis during this time. The uncertainty in the registration position is the distance traveled by the axis during this interval as shown by the equation. $\text{Uncertainty} = \text{Axis Speed} \left[\frac{\text{Position Units}}{\text{Second}}\right] \times \text{Delay}$ Use the formula given above to calculate the maximum registration position error for the expected axis speed. Alternatively, you can calculate the maximum axis speed for a specified registration accuracy by re-arranging this formula as shown. $\text{Maximum Speed} \left[\frac{\text{Position Units}}{\text{Second}}\right] = \frac{\text{Desired Accuracy [Position Units]}}{\text{Delay}}$
Registration 1 Time Registration 2 Time	AXIS_CONSUMED AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE AXIS_VIRTUAL	DINT	GSV Tag	Lower 32 bits of CST time The two Registration Time values contain the lower 32-bits of CST time at which their respective registration events occurred. Units for this attribute are in microseconds.

Attribute	Axis Type	Data Type	Access	Description
Resistive Brake Contact Delay	AXIS_SERVO_DRIVE	REAL	GSV SSV	Sec This attribute controls an optional external Resistive Brake Module (RBM). The RBM is between the drive and the motor and uses an internal contactor to switch the motor between the drive and a resistive load. The drive's RBM output controls this contactor. When the drive's RBM output is energized, the RBM contactor is switched from the load resistors to the UVW motor lines connecting the drive to the motor. This switching does not occur instantaneously and enabling the power structure too early can cause electrical arcing across the contactor. The resistive brake contact delay is the time needed to fully close the contactor across the UVW motor lines. To prevent electrical arcing across the contactor, the enabling of the drive's power structure is delayed. The delay time is variable depending on the RBM model. When applying an RBM, you must set the Resistive Brake Contact Delay to the recommended value found in the RBM specification. The following cases outline how the RBM output relates to the normal enable and disable sequences. **Case 1 – Enable Sequence** 1. Enable axis is initiated via MSO or MAH instruction. 2. Turn on RBM output to connect motor to drive. 3. Wait for Resistive Brake Contact Delay while RBM contacts close. 4. Drive power structure enabled (Drive Enable Status bit is set). 5. Turn on motor brake output to release brake. 6. Wait Brake Release Delay Time while motor brake releases. 7. Track Command reference (Servo Action Status bit is set). **Case 2 – Disable - Category 1 Stop** 1. Disable axis is initiated via an MSF instruction or a drive disable fault action. 2. Drive stops tracking command reference (Servo Action Status bit is cleared). 3. Apply Stopping Torque to stop motor. 4. Wait for zero speed or Stopping Time Limit. 5. Turn off brake output to engage motor brake. 6. Wait for Brake Engage delay while motor brake engages. 7. Disable drive power structure (Drive Enable Status bit is cleared). 8. Turn off RBM output to disconnect motor from drive. **Case 3 – Shutdown Category 0 Stop** 1. Drive stops tracking command reference (Servo Action Status bit is cleared). 2. Disable drive power structure (Drive Enable Status bit is cleared). 3. Turn off brake output to engage brake. 4. Turn off RBM output to disconnect motor from drive.
Rotary Axis	AXIS_CONSUMED AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE AXIS_VIRTUAL	SINT	GSV SSV* * Can only be set if axis is not enabled.	0 = Linear 1 = Rotary When the Rotary Axis attribute is set true (1), it lets the axis unwind. This gives infinite position range by unwinding the axis position whenever the axis moves through a complete physical revolution. The number of encoder counts per physical revolution of the axis is specified by the Position Unwind attribute. For Linear operation, the counts do not roll over. They are limited to +/- 2 billion.
Rotary Motor Inertia	AXIS_SERVO_DRIVE	REAL	SSV	The Rotary Motor Inertia attribute is a float that specifies the unloaded inertia of a rotary motor.

Attribute	Axis Type	Data Type	Access	Description
Rotary Motor Rated Speed	AXIS_SERVO_DRIVE	REAL	GSV	The Rotary Motor Rated Speed attribute is a float that specifies the nameplate rated speed of a rotary motor. For PM motors, this is generally specified at rated voltage based on rated current, rated torque, or rated power. For induction motors, this value is the speed of the motor driven at rated frequency under rated torque load. This value is synonymous with the term base speed.
Safe-Off Mode Active Status	AXIS_SERVO_DRIVE	BOOL	GSV Tag	This bit is the status indication of the Kinetix Drive's Safe-Off circuitry. If this bit is the following state: • ON - The Drive's Safety monitor circuitry encountered a loss of signal from Enable_1 or Enable_2. • OFF - The Drive's Safety monitor circuitry did not fault from Enable_1 or Enable_2. For the Kinetix Drive to pass back this status to the controller via this bit, the Drive must have firmware version 1.85 or higher.
SERCOS Error Code	AXIS_SERVO_DRIVE	INT	GSV* Tag	Error code returned by SERCOS module indicating source of drive parameter update failure. The SERCOS Error Code value can be used to identify the source of the drive parameter update failure that resulted in the Axis Configuration Fault.
SERCOS Fault	AXIS_SERVO_DRIVE	BOOL	Tag	Set when a requested SERCOS procedure fails to execute properly or the associated drive node detected a SERCOS communication fault.
SERCOS Ring Fault	AXIS_SERVO_DRIVE	BOOL	Tag	If this bit is set, there is a problem on the SERCOS ring; that is, the light is broken or a drive is powered down.
Servo Action Status	AXIS_CONSUMED AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE AXIS_VIRTUAL	BOOL	Tag	If this bit is: • ON — The axis is under servo control. • OFF — Servo action is disabled.
Servo Fault	AXIS_SERVO	DINT	Tag	Lets you access all servo fault bits in one 32-bit word. This tag is the same as the Servo Fault Bits attribute. **Servo Fault** — **Bit** Pos Soft Overtravel Fault — 0 Neg Soft Overtravel Fault — 1 Reserved — 2 Reserved — 3 Feedback Fault — 4 Feedback Noise Fault — 5 Reserved — 6 Reserved — 7 Positive Error Fault — 8 Drive Fault — 9 These fault bits are updated every base update period. Do you want any of these faults to give the controller a major fault? YES — Set the General Fault Type of the motion group = Major Fault. NO — You must write code to handle these faults.
Servo Fault Bits	AXIS_SERVO	DINT	GSV*	Lets you access all servo fault bits in one 32-bit word. This attribute is the same as the Servo Fault tag. **Servo Fault** — **Bit** Pos Soft Overtravel Fault — 0

Attribute	Axis Type	Data Type	Access	Description
				Neg Soft Overtravel Fault 1 Reserved 2 Reserved 3 Feedback Fault 4 Feedback Noise Fault 5 Reserved 6 Reserved 7 Positive Error Fault 8 Drive Fault 9 These fault bits are updated every base update period. Do you want any of these faults to give the controller a major fault? YES — Set the General Fault Type of the motion group = Major Fault. NO — You must write code to handle these faults.

Servo Feedback Type	AXIS_SERVO	SINT	GSV	This attribute provides a selection for the Feedback Type. 0 = A Quadrature B (AQB) 1 = Synchronous Serial Interface (SSI) 2 = Linear Displacement Transducer (LDT) **A Quadrature B Encoder Interface (AQB)** Servo modules, such as the 1756-M02AE, provide interface hardware to support incremental quadrature encoders equipped with standard 5-Volt differential encoder interface signals. This interface hardware provides a robust differential encoder input interface to condition each of the encoder signals before being applied to an Encoder-to-Digital Converter (EDC) FPGA. The EDC decodes the encoder signals and uses a 16-bit bidirectional counter to accumulate feedback counts. A regular Timer Event signal, applied to the EDC, latches the encoder counters for all axes simultaneously. This same Timer Event signal also triggers the servo interrupt service routine that performs the servo loop computations. One of the first things done by the interrupt service routine is to read the latched encoder counter values from the EDC. The change in the encoder counter value from the last timer event is computed and this delta value is added to a 32-bit signed integer position accumulator, which represents the Actual Position of the axis. The Actual Position value is used as feedback to the position servo loop and as input to the Watch Event Handler. The delta position value represents velocity feedback, which when configured to do so, may be filtered and applied to the inner velocity servo loop. **Synchronous Serial Interface (SSI)** Some servo modules, like the 1756-M02AS, provide an interface to transducers with Synchronous Serial Interface (SSI) outputs. SSI outputs use standard 5V differential signals (RS422) to transmit information from the transducer to the controller. The signals consist of a Clock generated by the controller and Data generated by the transducer. Each transducer with an SSI output provides output data of a specified number of bits of Binary or Gray code data. The controller must generate a stream of clock pulses with the correct number of bits and a frequency within the range supported by the transducer. The servo module can be configured via the Servo Axis Object to generate any number of clock pulses between 8 and 32, and the frequency can be set to 208kHz or 650kHz. The clock signal is maintained in the High state between pulse strings. The transducer shifts data out on the Data line MSB first on each rising edge of the clock signal. The transducer also maintains the data signal in specified states before and after the data is shifted out. These states are checked by the controller to detect missing transducers or broken wires. A Field Programmable Gate Array (FPGA) is used to implement a multi-channel SSI Interface on the controller. Each channel is functionally equivalent. **Linear Displacement Transducer (LDT)** Servo modules like the 1756-HYD02 use the Linear Magnetostrictive Displacement Transducer, or LDT. A Field Programmable Gate Array (FPGA) is used to implement a multi-channel LDT Interface. Each channel is functionally equivalent and is capable of interfacing to an LDT device with a maximum count of 240,000. The LDT interface includes transducer failure detection and digital filtering to reduce electrical noise. The FPGA can interface to two types of LDTs: Start/Stop and PWM. Start/Stop transducers accept an input (interrogate) signal to start the measurement cycle and respond with two pulses on the Return line. The time between the pulses is proportional to the position. PWM transducers respond to the interrogate signal with a long pulse on the Return line. The pulse width is proportional to the position. The FPGA generates the Interrogate signal every Servo Update time and measures the time between the Start/Stop pulses or the PWM pulse width. The resolution of the position measurement is determined by the frequency of the clock used for the time measurement. In the 1756-HYD02 design, a 60 MHz clock is used, and both edges of the clock signal are used for an effective time resolution of 8.3 nanoseconds. This translates into a position resolution better than 0.001 inch.

Attribute	Axis Type	Data Type	Access	Description
				Note: It is possible to achieve higher resolutions with PWM transducers that are configured to perform multiple internal measurements (recirculations) and report the sum of those measurements in the pulse width.
Servo Loop Configuration	AXIS_SERVO AXIS_SERVO_DRIVE	INT	GSV SSV	The Servo Loop Configuration attribute determines the configuration of the servo loop topology when the axis is set to 'servo'. 0 = custom 1 = feedback only 2 = aux. feedback only 3 = position servo 4 = aux. position servo 5 = dual position servo 6 = dual command servo 7 = aux. dual command servo 8 = velocity servo 9 = torque servo 10 = dual command/feedback servo
Servo Output Level	AXIS_SERVO	REAL	GSV Tag	**Important:** To use this attribute, choose it as one of the attributes for Real Time Axis Information for the axis. Otherwise, you won't see the right value as the axis runs. See Axis Info Select 1. Servo Output Level in Volts Servo Output Level is the current voltage level of the servo output of the specified axis. The Servo Output Level can be used in drilling applications, for example, where the servo module is interfaced to an external Torque Loop Servo Drive, to detect when the drill bit engages the surface of the work piece.
Servo Polarity Bits	AXIS_SERVO	DINT	GSV	0 = Feedback Polarity Negative 1 = Servo Polarity Negative **Feedback Polarity Negative** This Feedback Polarity Negative bit attribute controls the polarity of the encoder feedback and, when properly configured, insures that when the axis is moved in the user defined positive direction, and that the axis Actual Position value increases. This bit can be configured automatically using the MRHD and MAHD motion instructions. **Servo Polarity Negative** This Servo Polarity Negative bit attribute controls the polarity of the servo output to the drive. When properly configured along with the Feedback Polarity Negative bit, it insures that when the axis servo loop is closed, that it is closed as a negative feedback system and not an unstable positive feedback system. This bit can be configured automatically using the MRHD and MAHD motion instructions.

<table>
<tr><th>Attribute</th><th>Axis Type</th><th>Data Type</th><th>Access</th><th>Description</th></tr>
<tr><td>Servo Status</td><td>AXIS_SERVO</td><td>DINT</td><td>Tag</td><td>Allows access to the status bits for your servo loop in one 32-bit word. This tag is the same as the Servo Status Bits attribute.
<table>
<tr><th>Servo Status</th><th>Bit</th></tr>
<tr><td>Servo Action Status</td><td>0</td></tr>
<tr><td>Drive Enable Status</td><td>1</td></tr>
<tr><td>Shutdown Status</td><td>2</td></tr>
<tr><td>Process Status</td><td>3</td></tr>
<tr><td>Output Limit Status</td><td>4</td></tr>
<tr><td>Position Lock Status</td><td>5</td></tr>
<tr><td>Home Input Status</td><td>6</td></tr>
<tr><td>Reg 1 Input Status</td><td>7</td></tr>
<tr><td>Reg 2 Input Status</td><td>8</td></tr>
<tr><td>Resevered</td><td>9</td></tr>
<tr><td>Resevered</td><td>10</td></tr>
<tr><td>Drive Fault Input Status</td><td>11</td></tr>
</table></td></tr>
<tr><td>Servo Status Bits</td><td>AXIS_SERVO</td><td>DINT</td><td>GSV*</td><td>Allows access to the status bits for your servo loop in one 32-bit word. This attribute is the same as the Servo Status tag.
<table>
<tr><th>Servo Status</th><th>Bit</th></tr>
<tr><td>Servo Action Status</td><td>0</td></tr>
<tr><td>Drive Enable Status</td><td>1</td></tr>
<tr><td>Shutdown Status</td><td>2</td></tr>
<tr><td>Process Status</td><td>3</td></tr>
<tr><td>Output Limit Status</td><td>4</td></tr>
<tr><td>Position Lock Status</td><td>5</td></tr>
<tr><td>Home Input Status</td><td>6</td></tr>
<tr><td>Reg 1 Input Status</td><td>7</td></tr>
<tr><td>Reg 2 Input Status</td><td>8</td></tr>
<tr><td>Resevered</td><td>9</td></tr>
<tr><td>Resevered</td><td>10</td></tr>
<tr><td>Drive Fault Input Status</td><td>11</td></tr>
</table></td></tr>
<tr><td>Shutdown Status</td><td>AXIS_CONSUMED
AXIS_GENERIC
AXIS_SERVO
AXIS_SERVO_DRIVE
AXIS_VIRTUAL</td><td>BOOL</td><td>Tag</td><td>If this bit is:
• ON — The axis is in the Shutdown state.
• OFF — The axis is not in the Shutdown state.</td></tr>
</table>

Attribute	Axis Type	Data Type	Access	Description
Soft Overtravel Fault Action	AXIS_SERVO AXIS_SERVO_DRIVE	SINT	GSV SSV	<table><tr><th>Fault Action</th><th>Value</th></tr><tr><td>Shutdown</td><td>0</td></tr><tr><td>Disable Drive</td><td>1</td></tr><tr><td>Stop Motion</td><td>2</td></tr><tr><td>Status Only</td><td>3</td></tr></table> This controller attribute is replicated in the motion module.
SSI Clock Frequency	AXIS_SERVO	SINT	GSV	0 = 208 kHz 1 = 650 kHz This attribute provides for setting the Clock Frequency in kHz of the SSI device. This attribute is only active if the Transducer Type is set to SSI.
SSI Code Type	AXIS_SERVO	SINT	GSV	0 = Binary 1 = Gray This attribute provides for setting whether the SSI device is using Binary or Gray code. This attribute is only active if the Transducer Type is set to SSI.
SSI Data Length	AXIS_SERVO	SINT	GSV	This attribute provides for setting the data length of the SSI device. This attribute is only active if the Transducer Type is set to SSI.
Start Actual Position	AXIS_CONSUMED AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE AXIS_VIRTUAL	REAL	GSV Tag	Start Actual Position in Position Units Whenever a new motion planner instruction starts for an axis (for example, using an MAM instruction), the value of the axis command position and actual position is stored at the precise instant the motion begins. These values are stored as the Start Command Position and Start Actual Position respectively in the configured Position Units of the axis. Start Positions are useful to correct for any motion occurring between the *detection* of an event and the action *initiated* by the event. For instance, in coil winding applications, Start Command Positions can be used in an expression to compensate for overshooting the end of the bobbin before the gearing direction is reversed. If you know the position of the coil when the gearing direction was *supposed* to change, and the position at which it *actually* changed (the Start Command Position), you can calculate the amount of overshoot, and use it to correct the position of the wire guide relative to the bobbin.
Start Command Position	AXIS_CONSUMED AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE AXIS_VIRTUAL	REAL	GSV Tag	Start Command Position in Position Units Whenever a new motion planner instruction starts for an axis (for example, using an MAM instruction), the value of the axis command position and actual position is stored at the precise instant the motion begins. These values are stored as the Start Command Position and Start Actual Position respectively in the configured Position Units of the axis. Start Positions are useful to correct for any motion occurring between the detection of an event and the action initiated by the event. For instance, in coil winding applications, Start Command Positions can be used in an expression to compensate for overshooting the end of the bobbin before the gearing direction is reversed. If you know the position of the coil when the gearing direction was supposed to change, and the position at which it actually changed (the Start Command Position), you can calculate the amount of overshoot, and use it to correct the position of the wire guide relative to the bobbin.
Start Master Offset	AXIS_CONSUMED AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE AXIS_VIRTUAL	REAL	GSV Tag	Start Master Offset in Master Position Units The Start Master Offset is the position offset that was applied to the master side of the position cam when the last Motion Axis Move (MAM) instruction with the move type set to 'Absolute Master Offset' or 'Incremental Master Offset' was executed. The Start Master Offset is returned in master position units. The Start Master Offset shows the same unwind characteristic as the position of a linear axis.

Attribute	Axis Type	Data Type	Access	Description
Stopping Status	AXIS_CONSUMED AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE AXIS_VIRTUAL	BOOL	Tag	Set if there is a stopping process currently in progress. Cleared when the stopping process is complete. The stopping process is used to stop an axis (initiated by an MAS, MGS, Stop Motion fault action, or mode change).
Stopping Time Limit	AXIS_SERVO_DRIVE	REAL	GSV SSV	Sec This attribute maps directly to a SERCOS IDN. See the SERCOS Interface standard for a description. This attribute is automatically set. You usually do not need to change it.
Stopping Torque	AXIS_SERVO_DRIVE	REAL	GSV SSV	% Rated This attribute maps directly to a SERCOS IDN. See the SERCOS Interface standard for a description. This attribute is automatically set. You usually do not need to change it.
Strobe Actual Position	AXIS_CONSUMED AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE AXIS_VIRTUAL	REAL	GSV Tag	Strobe Actual Position in Position Units Strobe Actual Position and Strobe Command Position are used to simultaneously store a snap-shot of the actual, command position and master offset position of an axis when the MGSP (Motion Group Strobe Position) instruction is executed. The values are stored in the configured Position Units of the axis. Since the MGSP instruction simultaneously stores the actual and command positions for all axes in the specified group of axes, the resultant Strobe Actual Position and Strobe Command Position values can be used to perform real time calculations. For example, the Strobe Actual Positions can be compared between two axis to provide a form of 'slip compensation' in web handling applications.
Strobe Command Position	AXIS_CONSUMED AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE AXIS_VIRTUAL	REAL	GSV Tag	Strobe Command Position in Position Units Strobe Actual Position and Strobe Command Position are used to simultaneously store a snap-shot of the actual, command position and master offset position of an axis when the MGSP (Motion Group Strobe Position) instruction is executed. The values are stored in the configured Position Units of the axis. Since the MGSP instruction simultaneously stores the actual and command positions for all axes in the specified group of axes, the resultant Strobe Actual Position and Strobe Command Position values can be used to perform real time calculations. For example, the Strobe Actual Positions can be compared between two axis to provide a form of 'slip compensation' in web handling applications.
Strobe Master Offset	AXIS_CONSUMED AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE AXIS_VIRTUAL	REAL	GSV Tag	Strobe Master Offset in Master Position Units The Strobe Master Offset is the position offset that was applied to the master side of the position cam when the last Motion Group Strobe Position (MGSP) instruction was executed. The Strobe Master Offset is returned in master position units. The Strobe Master Offset shows the same unwind characteristic as the position of a linear axis.
Telegram Type	AXIS_SERVO_DRIVE	INT	GSV	Set to a value of 7, which means Application Telegram. See IDN 15 in IEC 1491.

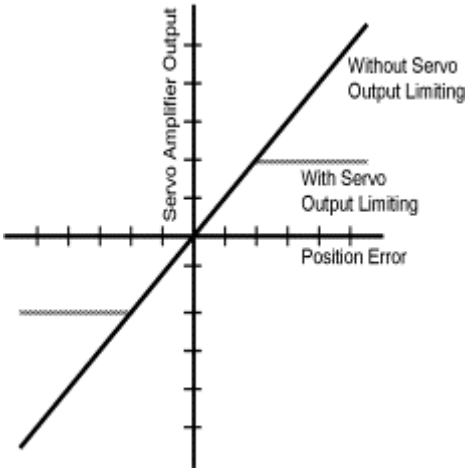
<table>
<tr><th>Attribute</th><th>Axis Type</th><th>Data Type</th><th>Access</th><th>Description</th></tr>
<tr><td>Test Direction Forward</td><td>AXIS_SERVO
AXIS_SERVO_DRIVE</td><td>SINT</td><td>GSV</td><td>The direction of axis travel during the last hookup test initiated by a MRHD (Motion Run Hookup Test) instruction.
0 = reverse
1 = forward (positive)<table><tr><th>For this Data type</th><th>Details</th></tr><tr><td>AXIS_SERVO</td><td>This value does not depend on the Servo Polarity Bits attribute. The MAHD (Motion Apply Hookup Test) instruction uses the Test Direction Forward attribute and the Test Output Polarity attribute to set the Servo Polarity Bits attribute for negative feedback and correct directional sense.</td></tr><tr><td>AXIS_SERVO_DRIVE</td><td>This value does not depend on the Drive Polarity attribute. The MAHD (Motion Apply Hookup Test) instruction uses the Test Direction Forward attribute and the Test Output Polarity attribute to set the Drive Polarity attribute for the correct directional sense.</td></tr></table></td></tr>
<tr><td>Test Increment</td><td>AXIS_SERVO
AXIS_SERVO_DRIVE</td><td>REAL</td><td>GSV
SSV</td><td>Position Units
The Motor Feedback Test Increment attribute is used in conjunction with the MRHD (Motion Run Hookup Diagnostic) instruction to determine the amount of motion that is necessary to satisfy the MRHD initiated test process. This value is typically set to approximately a quarter of a revolution of the motor.</td></tr>
<tr><td>Test Status</td><td>AXIS_SERVO
AXIS_SERVO_DRIVE</td><td>INT</td><td>GSV</td><td>0 = test process successful
1 = test in progress
2 = test process aborted by user
3 = test process time-out fault (~2 seconds)
4 = test failed – servo fault
5 = test failed – insufficient test increment
More for AXIS_SERVO_DRIVE data type…
6 = test failed – wrong polarity
7 = test failed – missing signal
8 = test failed – device comm error
9 = test failed – feedback config error
10 = test failed – motor wiring error
This attribute returns the status of the last run MRHD (Motion Run Hookup Diagnostic) instruction that initiates a hookup diagnostic process on the axis. Use this attribute to determine when the MRHD initiated operation successfully completed. Conditions may occur, however, that make it impossible to properly perform the operation. When that happens, the test process is automatically aborted and a test fault reported that is stored in the Test Status output parameter.</td></tr>
<tr><td>Time Cam Pending Status</td><td>AXIS_CONSUMED
AXIS_GENERIC
AXIS_SERVO
AXIS_SERVO_DRIVE
AXIS_VIRTUAL</td><td>BOOL</td><td>Tag</td><td>Set if a Time Cam motion profile is currently pending the completion of a currently executing cam profile. This would be initiated by executing an MATC instruction with Pending execution selected. This bit is cleared when the current time cam profile completes, initiating the start of the pending cam profile. This bit is also cleared if the time cam profile completes, or is superseded by some other motion operation.</td></tr>
</table>

Attribute	Axis Type	Data Type	Access	Description
Time Cam Status	AXIS_CONSUMED AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE AXIS_VIRTUAL	BOOL	Tag	Set if a Time Cam motion profile is currently in progress. Cleared when the Time Cam is complete or is superseded by some other motion operation.
Timer Event Fault	AXIS_SERVO AXIS_SERVO_DRIVE	BOOL	Tag	If this bit is set, there is an issue with the timer event that synchronizes the motion module's servo loop to the master timebase of the chassis (that is, Coordinated System Time). To clear this bit, reconfigure the motion module.
Torque Command	AXIS_SERVO_DRIVE	REAL	GSV Tag	**Important:** To use this attribute, choose it as one of the attributes for Real Time Axis Information for the axis. Otherwise, you won't see the right value as the axis runs. See Axis Info Select 1. %Rated The command when operating in Torque Mode in terms of % rated.

Torque Data Scaling	AXIS_SERVO_DRIVE	INT	GSV	This 16-bit attribute displays the scaling method to use on torque values (for example, Torque Command Value, and Bipolar torque limit value) with decimal values ranging from 0 to 127.
				Bit values are:
				• Bits 2-0: Scaling method
				  • 000 – percentage scaling
				  • 001 – linear scaling (force)
				  • 010 – rotational scaling (torque)
				• Bit 3:
				  • 0 – preferred scaling
				  • 1 – parameter scaling
				• Bit 4: Units
				  • 0 – Newton meter (Nm)
				  • 1 – inch pound force (lbf)
				• Bit 5: (reserved)
				• Bit 6: Data reference
				  • 0 – at the motor shaft
				  • 1 – at the load
				All other bits are reserved

Torque Data Scaling Exp	AXIS_SERVO_DRIVE	INT	GSV	This 16-bit unsigned attribute displays the scaling exponent for all torque data in a drive, with decimal values ranging from -2^15 to (2^15)-1.
				Bit values are:
				• Bit 14-0: Exponent value
				• Bit 15: Exponent sign:
				  • 0 - Positive
				  • 1 - Negative

Torque Data Scaling Factor	AXIS_SERVO_DRIVE	DINT	GSV	This 16-bit unsigned attribute displays the scaling factor for all torque data in a drive, with decimal values ranging from 1 to (2^16)-1.
Torque Feedback	AXIS_SERVO_DRIVE	REAL	GSV Tag	**Important:** To use this attribute, choose it as one of the attributes for Real Time Axis Information for the axis. Otherwise, you won't see the right value as the axis runs. See Axis Info Select 1. %Rated The torque feedback when operating in Torque Mode in terms of % rated.

Attribute	Axis Type	Data Type	Access	Description
Torque Limit Bipolar	AXIS_SERVO_DRIVE	REAL	GSV SSV	%Rated The Torque Limit attribute provides a method of limiting the maximum command current/torque to the motor to a specified level in terms of the motor's continuous current/torque rating. The output of the servo drive to the motor as a function of position servo error, with and without servo torque limiting, is shown below.  The torque limit specifies the maximum percentage of the motors rated current that the drive can command as positive or negative torque. For example, a torque limit of 150% shall limit the current delivered to the motor to 1.5 times the continuous current rating of the motor.
Torque Limit Negative	AXIS_SERVO_DRIVE	REAL	GSV SSV	%Rated This attribute maps directly to a SERCOS IDN. See the SERCOS Interface standard for a description. This attribute is automatically set. You usually do not need to change it.
Torque Limit Positive	AXIS_SERVO_DRIVE	REAL	GSV SSV	%Rated This attribute maps directly to a SERCOS IDN. See the SERCOS Interface standard for a description. This attribute is automatically set. You usually do not need to change it.
Torque Limit Source	AXIS_SERVO_DRIVE	DINT	GSV Tag	**Important:** To use this attribute, choose it as one of the attributes for Real Time Axis Information for the axis. Otherwise, you won't see the right value as the axis runs. See Axis Info Select 1. This parameter displays the present source (if any) of any torque limiting for the axis. 0 = Not Limited 1 = Neg. Torque Limit 2 = Pos. Torque Limit 3 = Amp Peak Limit 4 = Amp I(t) Limit 5 = Bus Regulator Limit 6 = Bipolar Torque Limit 7 = Motor Peak Limit 8 = Motor I(t) Limit 9 = Voltage Limit
Torque Limit Status	AXIS_SERVO_DRIVE	BOOL	Tag	Set when the magnitude of the axis torque command is greater than the configured Torque Limit.

Attribute	Axis Type	Data Type	Access	Description
Torque Offset	AXIS_SERVO AXIS_SERVO_DRIVE	REAL	GSV SSV Tag	Torque Offset from –100% to +100% Torque Offset compensation can be used to provide a dynamic torque command correction to the output of the velocity servo loop. Since this value is updated synchronously every Base Update Period, the Torque Offset can be tied into custom outer control loop algorithms using Function Block programming.
Torque Polarity	AXIS_SERVO_DRIVE	INT	GSV	G10/Configuration It maps directly to the SERCOS IDN. It is automatically set based on the current Drive Polarity Settings. All command bits are set according to the Command polarity bit value and all feedback bits are set according to the Feedback Polarity bit setting.
Torque Scaling	AXIS_SERVO AXIS_SERVO_DRIVE	REAL	GSV SSV	% / Position Units Per Second^2 This controller attribute is replicated in the motion module. The Torque Scaling attribute is used to convert the acceleration of the servo loop into equivalent % rated torque to the motor. This 'normalizes' the units of the servo loop's gain parameters so that their values are not affected by variations in feedback resolution, drive scaling, motor and load inertia, and mechanical gear ratios. In fact, the Torque Scaling value, when properly established, represents the inertia of the system and is related to the Tune Inertia attribute value by a factor of the Conversion Constant. • AXIS_SERVO — The Torque Scaling value is typically established by the MAAT instruction as part of the controller's automatic tuning procedure. • AXIS_SERVO_DRIVE — The Torque Scaling value is typically established by the drive's automatic tuning procedure. The value can be manually calculated, if necessary, using the following guidelines. Torque Scaling = 100% Rated Torque / (Acceleration @ 100% Rated Torque) For example, if this axis is using position units of motor revolutions (revs), and that with 100% rated torque applied to the motor, the motor accelerates at a rate of 3000 Revs/Sec^2, the Torque Scaling attribute value would be calculated as shown below. Torque Scaling = 100% Rated / (3000 RPS2) = 0.0333% Rated/ Revs Per Second^2 Note that if the Torque Scaling value does not reflect the true torque to acceleration characteristic of the system, the gains also do not reflect the true performance of the system.
Torque Threshold	AXIS_SERVO_DRIVE	REAL	GSV SSV	%Rated This attribute maps directly to a SERCOS IDN. See the SERCOS Interface standard for a description. This attribute is automatically set. You usually do not need to change it.
Torque Threshold Status	AXIS_SERVO_DRIVE	BOOL	Tag	Set when the magnitude of the physical axis Torque Feedback is less than the configured Torque Threshold.
Transform State Status	AXIS_CONSUMED AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE AXIS_VIRTUAL	BOOL	Tag	If the bit is: • ON — The axis is part of an active transform. • OFF — The axis is not part of an active transform.
Tune Acceleration	AXIS_SERVO AXIS_SERVO_DRIVE	REAL	GSV	Position Units / Sec^2 The Tune Acceleration and Tune Deceleration attributes return the measured acceleration and deceleration values for the last run tuning procedure. These values are used, in the case of an external torque servo drive configuration, to calculate the Tune Inertia value of the axis, and are also typically used by a subsequent MAAT (Motion Apply Axis Tune) to determine the tuned values for the Maximum Acceleration and Maximum Deceleration attributes.

Attribute	Axis Type	Data Type	Access	Description
Tune Acceleration Time	AXIS_SERVO AXIS_SERVO_DRIVE	REAL	GSV	Sec The Tune Acceleration Time and Tune Deceleration Time attributes return acceleration and deceleration time in seconds for the last run tuning procedure. These values are used to calculate the Tune Acceleration and Tune Deceleration attributes.
Tune Deceleration	AXIS_SERVO AXIS_SERVO_DRIVE	REAL	GSV	Position Units / Sec^2 The Tune Acceleration and Tune Deceleration attributes return the measured acceleration and deceleration values for the last run tuning procedure. These values are used, in the case of an external torque servo drive configuration, to calculate the Tune Inertia value of the axis, and are also typically used by a subsequent MAAT (Motion Apply Axis Tune) to determine the tuned values for the Maximum Acceleration and Maximum Deceleration attributes.
Tune Deceleration Time	AXIS_SERVO AXIS_SERVO_DRIVE	REAL	GSV	Sec The Tune Acceleration Time and Tune Deceleration Time attributes return acceleration and deceleration time in seconds for the last run tuning procedure. These values are used to calculate the Tune Acceleration and Tune Deceleration attributes.
Tune Inertia	AXIS_SERVO AXIS_SERVO_DRIVE	REAL	GSV	% / MegaCounts Per Sec^2 The Tune Inertia value represents the total inertia for the axis as calculated from the measurements made during the tuning procedure. In actuality, the units of Tune Inertia are not industry standard inertia units but rather in terms of percent (%) of rated drive output per MegaCounts/Sec^2 of feedback input. In this sense, it represents the input gain of torque servo drive. These units represent a more useful description of the inertia of the system as seen by the servo controller. The Tune Inertia value is used by the MAAT (Motion Apply Axis Tune) instruction to calculate the Torque Scaling. If the Tune Inertia value exceeds 100%Rated/MegaCounts Per $Second^2$, performance of the digital servo loop may be compromised due to excessive digitization noise associated with the velocity estimator. This noise is amplified by the Torque Scaling gain, which is related to the Tune Inertia factor and passed on to the torque output of the drive. A high Tune Inertia value can, thus, result in excitation of mechanical resonances and also result in excessive heating of the motor due to high torque ripple. The only solution to this problem is to lower the loop bandwidths and optionally apply some output filtering. Since the Tune Inertia value represents a measure of the true system inertia, this situation can occur when driving a high inertia load relative to the motor, that is, a high inertia mismatch. But it can also occur when working with a drive that is undersized for the motor or with a system having low feedback resolution. In general, the lower the Tune Inertia the better the performance of the digital servo loops approximates that of an analog servo system. The product of the Tune Inertia (% Rated/MCPS) and the Velocity Servo BW (Hertz) can be calculated to directly determine quantization noise levels. Based on this product, the tuning algorithm can take action to limit high frequency noise injection to the motor. For motors with a Tune Inertia BW product of 1000 or more, the LP Filter is applied with a Filter BW of 5x the Velocity Servo Bandwidth in Hertz. This limits the amount of phase lag introduced by the LP filter to ~12 degrees, which is relatively small compared to the 30 to 60 degrees of phase margin that exists for a typical tuned servo system. With a typical tuned LP filter BW value of 200 Hz, we can expect the high frequency quantization noise in the 1 KHz range to be attenuated roughly by a factor of 5. When the Tune Inertia BW product reaches 4000 or more, the LP filter alone is not going to be enough to manage the quantization noise level. So the tune algorithm begins to taper the system bandwidth by the ratio of 4000/(Tune Inertia * Vel Servo BW). This holds the quantization noise level at a fixed value, independent of the Tune Inertia BW product. For example, a motor with a Tune Inertia value of 213 and a Vel Servo BW of 41 Hz (8733 Inertia BW product) tunes with a Pos P Gain of 46 and a Vel P Gain of 117 and LP Filter BW of 93. This is a good noise-free gain set.

Attribute	Axis Type	Data Type	Access	Description
Tune Rise Time	AXIS_SERVO	REAL	GSV	Seconds The Tune Rise Time attribute returns the axis rise time as measured during the tuning procedure. This value is only applicable to axes configured for interface to an external velocity servo drive. In this case, the Tune Rise Time attribute value is used to calculate the Tune Velocity Bandwidth.
Tune Speed Scaling	AXIS_SERVO	REAL	GSV	% / KiloCounts Per Seconds The Tune Speed Scaling attribute returns the axis drive scaling factor measured during the tuning procedure. This value is only applicable to axes configured for interface to an external velocity servo drive. In this case, the Tune Speed Scaling attribute value is directly applied to the Velocity Scaling attribute by a subsequent MAAT (Motion Apply Axis Tune) instruction.
Tune Status	AXIS_SERVO AXIS_SERVO_DRIVE	INT	GSV	0 = tune process successful 1 = tune in progress 2 = tune process aborted by user 3 = tune process timed out 4 = AXIS_SERVO — tune process failed due to servo fault = AXIS_SERVO_DRIVE — tune process failed due to drive fault 5 = axis reached Tuning Travel Limit 6 = axis polarity set incorrectly **More codes for a AXIS_SERVO_DRIVE** 7 = tune measurement fault 8 = tune configuration fault The Tune Status attribute returns status of the last run MRAT (Motion Run Axis Tuning) instruction that initiates a tuning procedure on the targeted axis. Use the attribute to determine when the MRAT initiated operation successfully completed. Conditions may occur, however, that make it impossible for the control to properly perform the operation. When this is the case, the tune process is automatically aborted and a tune fault reported that is stored in the Tune Status output parameter.

Attribute	Axis Type	Data Type	Access	Description
Tuning Configuration Bits	AXIS_SERVO AXIS_SERVO_DRIVE	DINT	GSV SSV	Bits 0 = Tuning Direction Reverse 1 = Tune Position Error Integrator 2 = Tune Velocity Error Integrator 3 = Tune Velocity Feedforward 4 = Tune Acceleration Feedforward 5 = Tune Output Low-Pass Filter 6 = bidirectional Tuning 7 = Tune Backlash Compensation 8 = Tune Torque Offset **Tuning Direction Reverse** The Tune Direction Reverse bit determines the direction of the tuning procedure. If this bit is set (true), motion is initiated in the reverse (or negative) direction. **Tune Position Error Integrator** If this bit is: • ON — The tuning procedure calculates the Position Integral Gain. • OFF — The tuning procedure sets the Position Integral Gain to 0. **Tune Velocity Error Integrator** If this bit is: • ON — The tuning procedure calculates the Velocity Integral Gain. • OFF — The tuning procedure sets the Velocity Integral Gain to 0. **Tune Velocity Feedforward** If this bit is: • ON — The tuning procedure calculates the Velocity Feedforward Gain. • OFF — The tuning procedure sets the Velocity Feedforward Gain to 0. **Tune Acceleration Feedforward** If this bit is: • ON — The tuning procedure calculates the Acceleration Feedforward Gain. • OFF — The tuning procedure sets the Acceleration Feedforward Gain to 0. **Tune Output Low-Pass Filter** If this bit is: • ON — The tuning procedure calculates the Output Filter Bandwidth. • OFF — The tuning procedure sets the Output Filter Bandwidth to 0, which disables the filter.

Attribute	Axis Type	Data Type	Access	Description
Tuning Configuration Bits (continued)				**Bidirectional Tuning** The Bidirectional Tuning bit determines whether the tuning procedure is unidirectional or bidirectional. If this bit is set (true), the tuning motion profile is first initiated in the specified tuning direction and then is repeated in the opposite direction. Information returned by the Bidirectional Tuning profile can be used to tune Backlash Compensation and Torque Offset. When configured for a hydraulics External Drive Type, the bidirectional tuning algorithm also computes the Directional Scaling Ratio. **Tune Backlash Compensation** This tuning configuration is only valid if configured for bidirectional tuning. If this bit is: • ON — The tuning procedure calculates the Backlash Compensation Gain. • OFF — The Backlash Compensation Gain is not affected. **Tune Torque Offset** This tuning configuration is only valid if configured for bidirectional tuning. If this bit is: • ON — The tuning procedure calculates the Torque Offset. • OFF — The Torque Offset is not affected.
Tuning Speed	AXIS_SERVO AXIS_SERVO_DRIVE	REAL	GSV SSV	Position Units / Seconds The Tuning Speed attribute sets the maximum speed of the tuning procedure. This attribute should be set to the desired maximum operating speed of the motor before you run the tuning procedure. The tuning procedure measures maximum acceleration and deceleration rates based on ramps to and from the Tuning Speed. Thus, the accuracy of the measured acceleration and deceleration capability is reduced by tuning at a speed other than the desired operating speed of the system.
Tuning Torque	AXIS_SERVO AXIS_SERVO_DRIVE	REAL	GSV SSV	% The Tuning Torque attribute determines the maximum torque of the tuning procedure. This attribute should be set to the desired maximum safe torque level before you run the tuning procedure. The default value is 100%, which yields the most accurate measure of the acceleration and deceleration capabilities of the system. In some cases, a lower tuning torque limit value may be desirable to limit the stress on the mechanics during the tuning procedure. In this case, the acceleration and deceleration capabilities of the system are extrapolated based on the ratio of the tuning torque to the maximum torque output of the system. Note that the extrapolation error increases as the Tuning Torque value decreases.
Tuning Travel Limit	AXIS_SERVO AXIS_SERVO_DRIVE	REAL	GSV SSV	Position Units The Tuning Travel Limit attribute limits the travel of the axis during the tuning procedure. If the axis cannot complete the tuning procedure before exceeding the Tuning Travel Limit, the motion module stops the tuning procedure and reports that the Tuning Travel Limit was exceeded via the Tune Status attribute. This does not mean that the Tuning Travel Limit was actually exceeded, but that had the tuning procedure gone to completion, the limit would be exceeded.
Velocity Command	AXIS_SERVO AXIS_SERVO_DRIVE	REAL	GSV Tag	**Important:** To use this attribute, choose it as one of the attributes for Real Time Axis Information for the axis. Otherwise, you won't see the right value as the axis runs. See Axis Info Select 1. Velocity Command in Position Units / Seconds Velocity Command is the current velocity reference to the velocity servo loop, in the configured axis Position Units per Second, for the specified axis. The Velocity Command value, hence, represents the output of the outer position control loop. Velocity Command is not to be confused with Command Velocity, which represents the rate of change of Command Position input to the position servo loop.
Velocity Data Scaling	AXIS_SERVO_DRIVE	INT	GSV	This attribute is derived from the Drive Units attribute. See IDN 44 in IEC 1491.

Attribute	Axis Type	Data Type	Access	Description
Velocity Data Scaling Exp	AXIS_SERVO_DRIVE	INT	GSV	This attribute is derived from the Drive Units attribute. See IDN 46 in IEC 1491.
Velocity Data Scaling Factor	AXIS_SERVO_DRIVE	DINT	GSV	This attribute is derived from the Drive Units attribute. See IDN 45 in IEC 1491.
Velocity Droop	AXIS_SERVO_DRIVE	REAL	GSV SSV	Position Units / sec This attribute maps directly to a SERCOS IDN. See the SERCOS Interface standard for a description. This attribute is automatically set. You usually do not need to change it.
Velocity Error	AXIS_SERVO AXIS_SERVO_DRIVE	REAL	GSV Tag	**Important:** To use this attribute, choose it as one of the attributes for Real Time Axis Information for the axis. Otherwise, you won't see the right value as the axis runs. See Axis Info Select 1. Velocity Error in Position Units / Seconds Velocity Error is the difference, in configured axis Position Units per Second, between the commanded and actual velocity of an axis. For an axis with an active velocity servo loop, velocity error is used, along with other error terms, to drive the motor to the condition where the velocity feedback is equal to the velocity command..
Velocity Feedback	AXIS_SERVO AXIS_SERVO_DRIVE	REAL	GSV Tag	**Important:** To use this attribute, choose it as one of the attributes for Real Time Axis Information for the axis. Otherwise, you won't see the right value as the axis runs. See Axis Info Select 1. Velocity Feedback in Position Units / Seconds Velocity Feedback is the actual velocity of the axis as estimated by the motion module, in the configured axis Position Units per second. The estimated velocity is computed by applying a 1 KHz low-pass filter to the change in actual position over the servo update interval. Velocity Feedback is a signed value—the sign (+ or -) depends on which direction the axis is currently moving.

Attribute	Axis Type	Data Type	Access	Description
Velocity Feedforward Gain	AXIS_SERVO AXIS_SERVO_DRIVE	REAL	GSV SSV	% This controller attribute is replicated in the motion module. Servo Drives require non-zero command input to generate steady-state axis acceleration or velocity. To provide the non-zero output from the Servo Module, a non-zero position or velocity error needs to be present. We call this dynamic error while moving 'following error'. Well, this non-zero following error condition is a situation we are trying to avoid. You ideally want zero following the error. This could be achieved through use of the position integral gain controls as described above, but typically the response time of the integrator action is too slow to be effective. An alternative approach with superior dynamic response is to use Velocity and Acceleration Feedforward. The Velocity Feedforward Gain attribute is used to provide the Velocity Command output necessary to generate the commanded velocity. It does this by scaling the current Command Velocity by the Velocity Feedforward Gain and adding it as an offset to the Velocity Command generated by the position loop control elements. With this done, the position loop control elements do not need to generate much of a contribution to the Velocity Command, hence, the Position Error value is significantly reduced. Hence, the Velocity Feedforward Gain allows the following error of the servo system to be reduced to nearly zero when running at a constant speed. This is important in applications such as electronic gearing and synchronization applications where it is necessary that the actual axis position not significantly lag behind the commanded position at any time. The optimal value for Velocity Feedforward Gain is 100% theoretically. In reality, however, the value may need to be tweaked to accommodate velocity loops with non-infinite loop gain and other application considerations. One thing that may force a smaller Velocity Feedforward value is that increasing amounts of feedforward tends to exacerbate axis overshoot. If necessary, the Velocity Feedforward Gain may be 'tweaked' from the 100% value by running a simple user program that jogs the axis in the positive direction and monitor the Position Error of the axis during the jog. Increase the Velocity Feedforward Gain until the Position Error at constant speed is as small as possible, but still positive. If the Position Error at constant speed is negative, the actual position of the axis is *ahead* of the command position. If this occurs, decrease the Velocity Feedforward Gain such that the Position Error is again positive. Note that reasonable maximum velocity, acceleration, and deceleration values must be entered to jog the axis.

Attribute	Axis Type	Data Type	Access	Description
Velocity Integral Gain	AXIS_SERVO AXIS_SERVO_DRIVE	REAL	GSV SSV	1/mSeconds-Seconds This controller attribute is replicated in the motion module. When configured for a torque (current) loop servo drive, every servo update, the current Velocity Error is also accumulated in a variable called the Velocity Integral Error. This value is multiplied by the Velocity Integral Gain to produce a component to the Servo Output or Torque Command that attempts to correct for the velocity error. The characteristic of Vel I Gain correction, however, is that any non-zero Velocity Error accumulates in time to generate enough force to make the correction. This attribute of Vel I Gain makes it invaluable in applications where velocity accuracy is critical. The higher the Vel I Gain value the faster the axis is driven to the zero Velocity Error condition. Unfortunately, I Gain control is intrinsically unstable. Too much I Gain results in axis oscillation and servo instability. In certain cases, Vel I Gain control is disabled. One such case is when the servo output to the axis' drive is saturated. Continuing integral control behavior in this case would only exacerbate the situation. Another common case is when performing certain motion. When the Integrator Hold Enable attribute is set, the servo loop automatically disables the integrator during commanded motion. Due to the destabilizing nature of Integral Gain, it is recommended that Position Integral Gain and Velocity Integral Gain be considered mutually exclusive. If Integral Gain is needed for the application, use one or the other. In general, where static positioning accuracy is required, Velocity Integral Gain is the better choice. The typical value for the Velocity Integral Gain is ~15 $mSec^{-1}$-Sec^{-1}. **If you have an AXIS_SERVO_DRIVE data type…** While the Vel I Gain, if employed, is typically established by the automatic servo tuning procedure, the Pos I Gain value may also be set manually. Before doing this, it must be stressed that the Torque Scaling factor for the axis must be established for the drive system. See Torque Scaling attribute description for an explanation of how the Torque Scaling factor can be calculated. Once this is done, the Vel I Gain can be computed based on the current or computed value for the Vel P Gain using the following formula: Vel I Gain = 0.25 * 0.001 Sec/mSec * (Vel P Gain)^2 Assuming a Vel P Gain value of 0.25 Sec^{-1}, this results in a Vel I Gain value of ~15.6 $mSec^{-1}$-Sec^{-1}-
Velocity Integrator Error	AXIS_SERVO AXIS_SERVO_DRIVE	REAL	GSV Tag	**Important:** To use this attribute, choose it as one of the attributes for Real Time Axis Information for the axis. Otherwise, you won't see the right value as the axis runs. See Axis Info Select 1. Velocity Integrator Error in Position Units – mSec / Sec Velocity Integrator Error is the running sum of the Velocity Error, in the configured axis Position Units per Second, for the specified axis. For an axis with an active velocity servo loop, the velocity integrator error is used, along with other error terms, to drive the motor to the condition where the velocity feedback is equal to the velocity command.
Velocity Limit Bipolar	AXIS_SERVO_DRIVE	REAL	GSV SSV	Position Units / sec This attribute maps directly to a SERCOS IDN. See the SERCOS Interface standard for a description. This attribute is automatically set. You usually do not need to change it.
Velocity Limit Negative	AXIS_SERVO_DRIVE	REAL	GSV SSV	Position Units / sec This attribute maps directly to a SERCOS IDN. See the SERCOS Interface standard for a description. This attribute is automatically set. You usually do not need to change it.
Velocity Limit Positive	AXIS_SERVO_DRIVE	REAL	GSV SSV	Position Units / sec This attribute maps directly to a SERCOS IDN. See the SERCOS Interface standard for a description. This attribute is automatically set. You usually do not need to change it.
Velocity Limit Status	AXIS_SERVO_DRIVE	BOOL	Tag	Set when the magnitude of the commanded velocity to the velocity servo loop input is greater than the configured Velocity Limit.

Attribute	Axis Type	Data Type	Access	Description
Velocity Lock Status	AXIS_SERVO_DRIVE	BOOL	Tag	Set when the magnitude of the physical axis Velocity Feedback is within the configured Velocity Window of the current velocity command.
Velocity Offset	AXIS_SERVO AXIS_SERVO_DRIVE	REAL	GSV SSV Tag	Velocity Offset in Position Units / Sec Velocity Offset compensation can be used to give a dynamic velocity correction to the output of the position servo loop. Since this value is updated synchronously every Base Update Period, the Velocity Offset can be tied into custom outer control loop algorithms using Function Block programming.
Velocity Polarity	AXIS_SERVO_DRIVE	INT	GSV	This attribute is derived from the Drive Polarity attribute. See IDN 42 in IEC 1491.
Velocity Proportional Gain	AXIS_SERVO AXIS_SERVO_DRIVE	REAL	GSV SSV	1/Seconds This controller attribute is replicated in the motion module. **AXIS_SERVO** When configured for a torque (current) loop servo drive, the servo module's digital velocity loop provides damping without the requirement for an analog tachometer. The Velocity Error is multiplied by the Velocity Proportional Gain to produce a component to the Servo Output or Torque Command that ultimately attempts to correct for the velocity error, creating the damping effect. Thus, increasing the Velocity Proportional Gain results in smoother motion, enhanced acceleration, reduced overshoot, and greater system stability. The velocity loop also allows higher effective position loop gain values to be used, however, too much Velocity Proportional Gain leads to high frequency instability and resonance effects. Note that units for Velocity Proportional Gain are identical to that of the Position Proportional Gain making it easy to perform classic inches/min/mil calculations to determine static stiffness or damping. **Maximum Bandwidth** There are limitations to the maximum bandwidth that can be achieved for the velocity loop based on the dynamics of the torque loop of the servo drive and the desired damping of the system, Z. These limitations may be expressed as follows: Bandwidth (Velocity) = $0.25 * 1/Z^2 *$ Bandwidth (Torque) For example, if the bandwidth of the drive's torque loop is 100 Hz and the damping factor, Z, is 0.8, the velocity bandwidth is approximately 40 Hz. Based on this number, the corresponding gains for the loop can be computed. Note that the bandwidth of the torque loop includes feedback sampling delay and filter time constant. The velocity loop in the motion controller is not used when the servo module is configured for a velocity loop servo drive, Thus, establishing the Velocity Proportional Gain is not required in this case. The typical value for the Velocity Proportional Gain is ~250 Sec^{-1}.

Attribute	Axis Type	Data Type	Access	Description
Velocity Proportional Gain (continued)				**AXIS_SERVO_DRIVE** The standard RA SERCOS drive's digital velocity loop provides damping without the requirement for an analog tachometer. The Velocity Error is multiplied by the Velocity Proportional Gain to produce a Torque Command that ultimately attempts to correct for the velocity error, creating the damping effect. Thus, increasing the Velocity Proportional Gain results in smoother motion, enhanced acceleration, reduced overshoot, and greater system stability. The velocity loop also allows higher effective position loop gain values to be used, however, too much Velocity Proportional Gain leads to high frequency instability and resonance effects. Note that units for Velocity Proportional Gain are identical to that of the Position Proportional Gain making it easy to perform classic calculations to determine damping and bandwidth. If you know the desired unity gain bandwidth of the velocity servo in Hertz, use the following formula to calculate the corresponding P gain. Vel P Gain = Bandwidth (Hertz) / 6.28 In general, modern velocity servo systems typically run with a unity gain bandwidth of ~40 Hertz. The typical value for the Velocity Proportional Gain is ~250 Sec^{-1}. **Maximum Bandwidth** There are limitations to the maximum bandwidth that can be achieved for the velocity loop based on the dynamics of the inner torque loop of the system and the desired damping of the system, Z. These limitations may be expressed as follows: Bandwidth (Velocity) = 0.25 * $1/Z^2$ * Bandwidth (Torque) For example, if the bandwidth of the drive's torque loop is 100 Hz and the damping factor, Z, is 0.8, the velocity bandwidth is approximately 40 Hz. Based on this number, the corresponding gains for the loop can be computed. Note that the bandwidth of the torque loop includes feedback sampling delay and filter time constant.
Velocity Scaling	AXIS_SERVO	REAL	GSV SSV	% / Position Units Per Second This controller attribute is replicated in the motion module. The Velocity Scaling attribute is used to convert the output of the servo loop into equivalent voltage to an external velocity servo drive. This 'normalizes' the units of the servo loop gain parameters so that their values are not affected by variations in feedback resolution, drive scaling, or mechanical gear ratios. The Velocity Scaling value is typically established by servo's automatic tuning procedure but these values can be calculated, if necessary, using the following guidelines. If the axis is using a velocity servo drive, the software velocity loop in the servo module is disabled. In this case, the Velocity Scaling value can be calculated by the following formula: Velocity Scaling = 100% / (Speed @ 100%) For example, if this axis is using position units of motor revolutions (revs), and the servo drive is scaled such that with an input of 100% (for example, 10 Volts) the motor goes 5,000 RPM (or 83.3 RPS), the Torque Scaling attribute value would be calculated as shown below. Velocity Scaling = 100% / (83.3 RPS) = 1.2% / Revs Per Second

<table>
<tr><th>Attribute</th><th>Axis Type</th><th>Data Type</th><th>Access</th><th>Description</th></tr>
<tr><td>Velocity Servo Bandwidth</td><td>AXIS_SERVO
AXIS_SERVO_DRIVE</td><td>REAL</td><td>GSV
SSV</td><td>Hertz
The value for the Velocity Servo Bandwidth represents the unity gain bandwidth that is to be used to calculate the gains for a subsequent MAAT (Motion Apply Axis Tune) instruction. The unity gain bandwidth is the frequency beyond which the velocity servo is unable to provide any significant position disturbance correction. In general, within the constraints of a stable servo system, the higher the Velocity Servo Bandwidth is the better the dynamic performance of the system. A maximum value for the Velocity Servo Bandwidth is generated by the MRAT (Motion Run Axis Tune) instruction. Computing gains based on this maximum value via the MAAT instruction results in dynamic response in keeping with the current value of the Damping Factor described above. Alternatively, the responsiveness of the system can be ‘softened’ by reducing the value of the Velocity Servo Bandwidth before executing the MAAT instruction..
There are practical limitations to the maximum Velocity Servo Bandwidth for the velocity servo loop based on the drive system and, in some cases, the desired damping factor of the system, Z. Exceeding these limits could result in an unstable servo operation.
<table><tr><th>Data type</th><th>Bandwidth limits</th></tr><tr><td>AXIS_SERVO</td><td>For an external velocity loop servo drive,
Max Velocity Servo Bandwidth (Hz) = 0.159 * 2/Tune Ris Time
For an external torque loop servo drive,
Max Velocity Servo Bandwidth (Hz) = 0.159 * 0.25 * 1/2 * 1/Drive Model Time Constant</td></tr><tr><td>AXIS_SERVO_DRIVE</td><td>Max Velocity Servo Bandwidth (Hz) = 0.159 * 0.25 * 1/2² * 1/Drive Model Time Constant</td></tr></table>The factor of 0.159 represents the 1/2PI factor required to convert Radians per Second units to Hertz.</td></tr>
<tr><td>Velocity Standstill Status</td><td>AXIS_SERVO_DRIVE</td><td>BOOL</td><td>Tag</td><td>Set when the magnitude of the physical axis Velocity Feedback is less than the configured Velocity Standstill Window.</td></tr>
<tr><td>Velocity Standstill Window</td><td>AXIS_SERVO_DRIVE</td><td>REAL</td><td>GSV
SSV</td><td>Position Units / sec
This attribute maps directly to a SERCOS IDN. See the SERCOS Interface standard for a description. This attribute is automatically set. You usually do not need to change it.</td></tr>
<tr><td>Velocity Threshold</td><td>AXIS_SERVO_DRIVE</td><td>REAL</td><td>GSV
SSV</td><td>Position Units / sec
This attribute maps directly to a SERCOS IDN. See the SERCOS Interface standard for a description. This attribute is automatically set. You usually do not need to change it.</td></tr>
<tr><td>Velocity Threshold Status</td><td>AXIS_SERVO_DRIVE</td><td>BOOL</td><td>Tag</td><td>Set when the magnitude of the physical axis Velocity Feedback is less than the configured Velocity Threshold.</td></tr>
<tr><td>Velocity Window</td><td>AXIS_SERVO_DRIVE</td><td>REAL</td><td>GSV
SSV</td><td>Position Units / sec
This attribute maps directly to a SERCOS IDN. See the SERCOS Interface standard for a description. This attribute is automatically set. You usually do not need to change it.</td></tr>
<tr><td>Watch Event Armed Status</td><td>AXIS_CONSUMED
AXIS_GENERIC
AXIS_SERVO
AXIS_SERVO_DRIVE
AXIS_VIRTUAL</td><td>BOOL</td><td>Tag</td><td>Set when a watch event is armed through execution of the MAW (Motion Arm Watch) instruction. Cleared when a watch event occurs or an MDW (Motion Disarm Watch) instruction is executed.</td></tr>
</table>

Attribute	Axis Type	Data Type	Access	Description
Watch Event Status	AXIS_CONSUMED AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE AXIS_VIRTUAL	BOOL	Tag	Set when a watch event occurs. Cleared when another MAW (Motion Arm Watch) instruction or an MDW (Motion Disarm Watch) instruction is executed.
Watch Event Task	AXIS_CONSUMED AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE AXIS_VIRTUAL	DINT	MSG	Shows which task is triggered when the watch event happens. • An instance of 0 means that no event task is configured to be triggered by the watch event. • The triggering of the task occurs simultaneously with the setting of the Process Complete bit for the instruction that armed the watch event. • The controller sets this attribute. Do not set it by an external device.
Watch Position	AXIS_CONSUMED AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE AXIS_VIRTUAL	REAL	GSV Tag	Watch Position in Position Units Watch Position is the current set-point position of an axis, in the configured axis Position Units, as configured in the last, most recently executed, MAW (Motion Arm Watch) instruction for that axis.

Additional error code information

See these manuals for more information about error codes displayed on drives and/or multi-axis motion control systems.

Publication	Description
Kinetix 2000 Multi-Axis Servo Drive User Manual, publication 2093-UM001	Provides detailed installation instructions for mounting, wiring, and troubleshooting your Kinetix 2000 drive, and system integration for your drive/motor combination with a Logix controller.
Kinetix 6000 Multi-Axis Drives User Manual, publication 2094-UM001	Provides detailed installation instructions for mounting, wiring, and troubleshooting your Kinetix 6000 drive, and system integration for your drive/motor combination with a Logix controller.
Kinetix 7000 High Power Servo Drive User Manual, publication 2099-UM001	Provides detailed installation instructions for mounting, wiring, and troubleshooting your Kinetix 7000 drive, and system integration for your drive/motor combination with a Logix controller.
Ultra 3000 Digital Servo Drive Installation Instructions, publication 2098-IN003	Provides the mounting, wiring, and connecting procedures for the Ultra3000 and standard Rockwell Automation/Allen-Bradley motors recommended for use with the Ultra3000.
8720 High Performance Drive Installation Instructions, publication 8720MC-IN001	Provides the mounting, wiring, and connecting procedures for the 8720MC and standard Rockwell Automation/ Allen-Bradley motors recommended for use with the 8720MC.

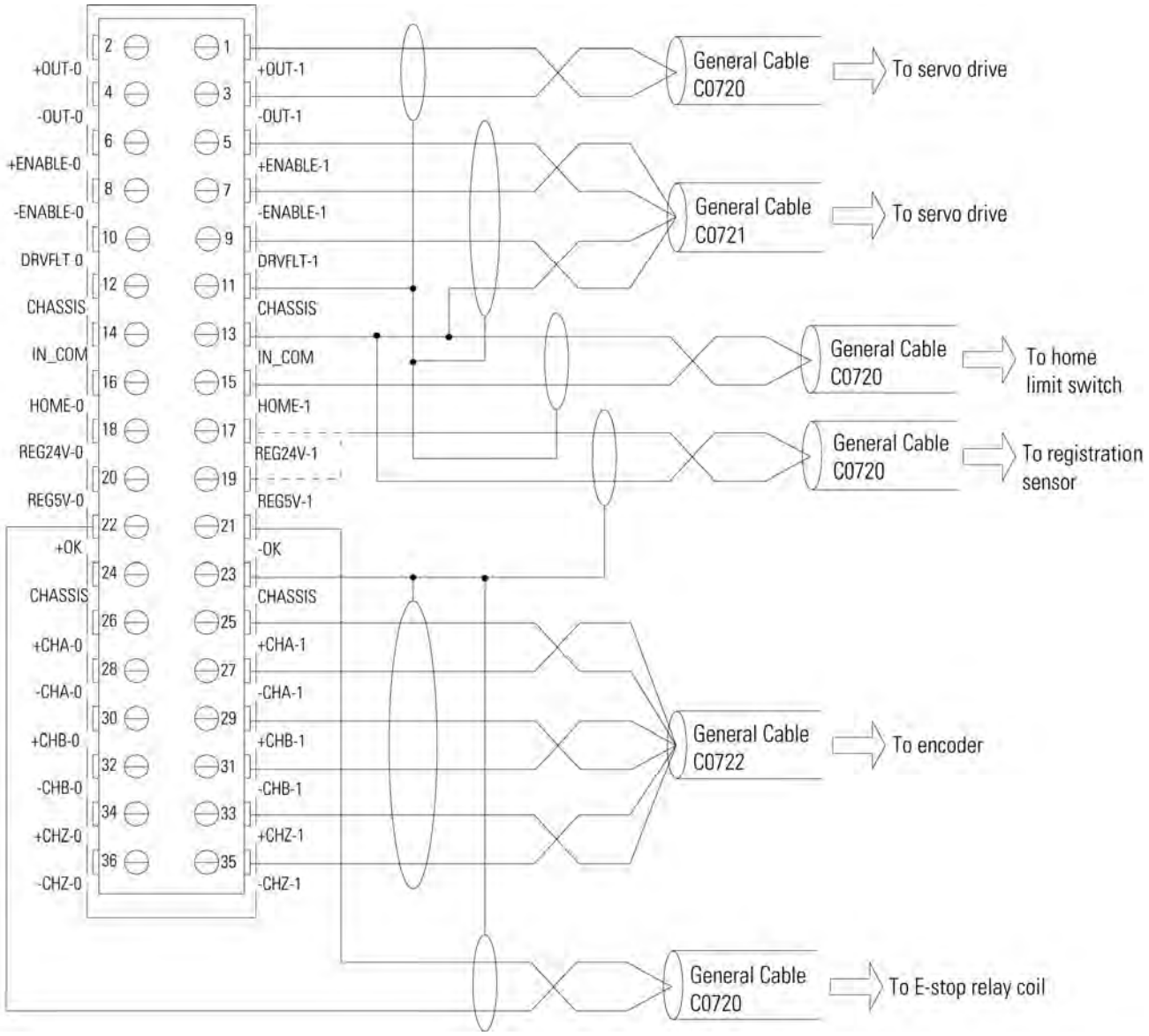
Wiring diagrams

Introduction for Wiring Diagrams

Use the diagrams in this appendix to wire the motion control equipment of your control system.

1756-M02AE module

The following image illustrates the 1756-M02AE module.

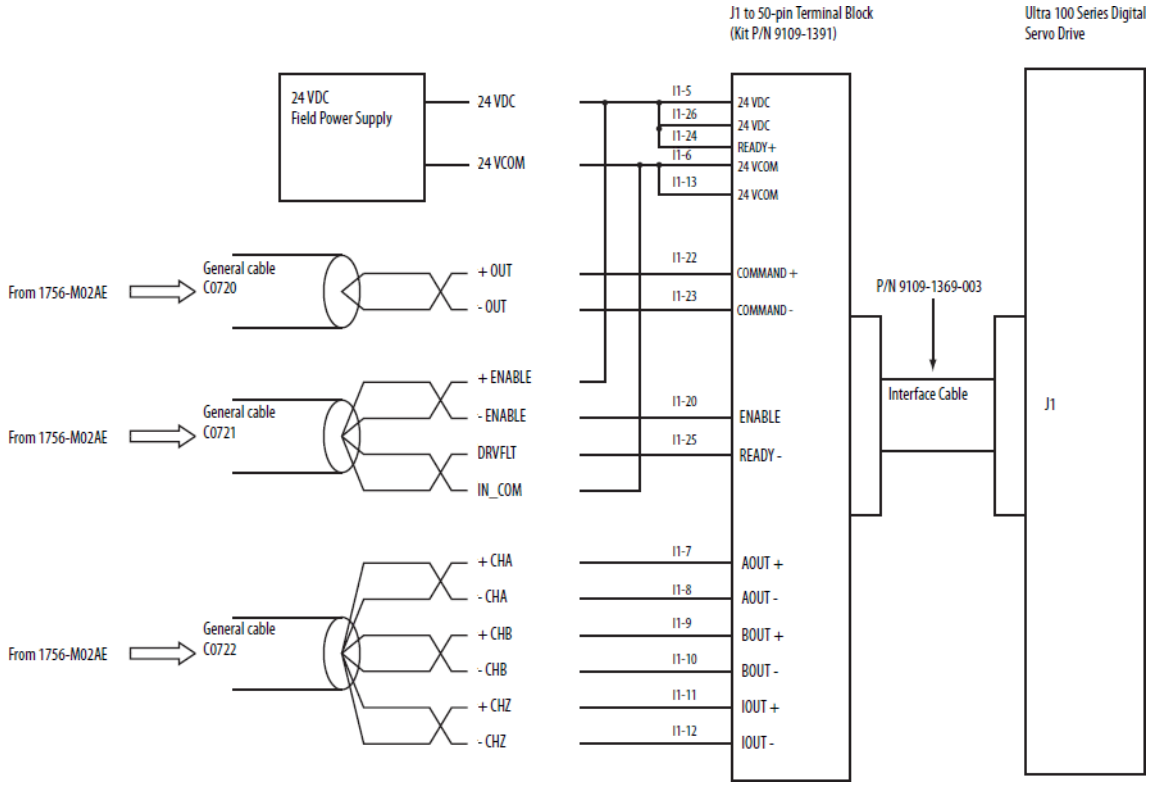


Notes

The example shows the wiring for Axis 1 Wire Axis 0 the same way.

Ultra 100 Series Drive

The following image illustrates the Ultra 1000 series drive.

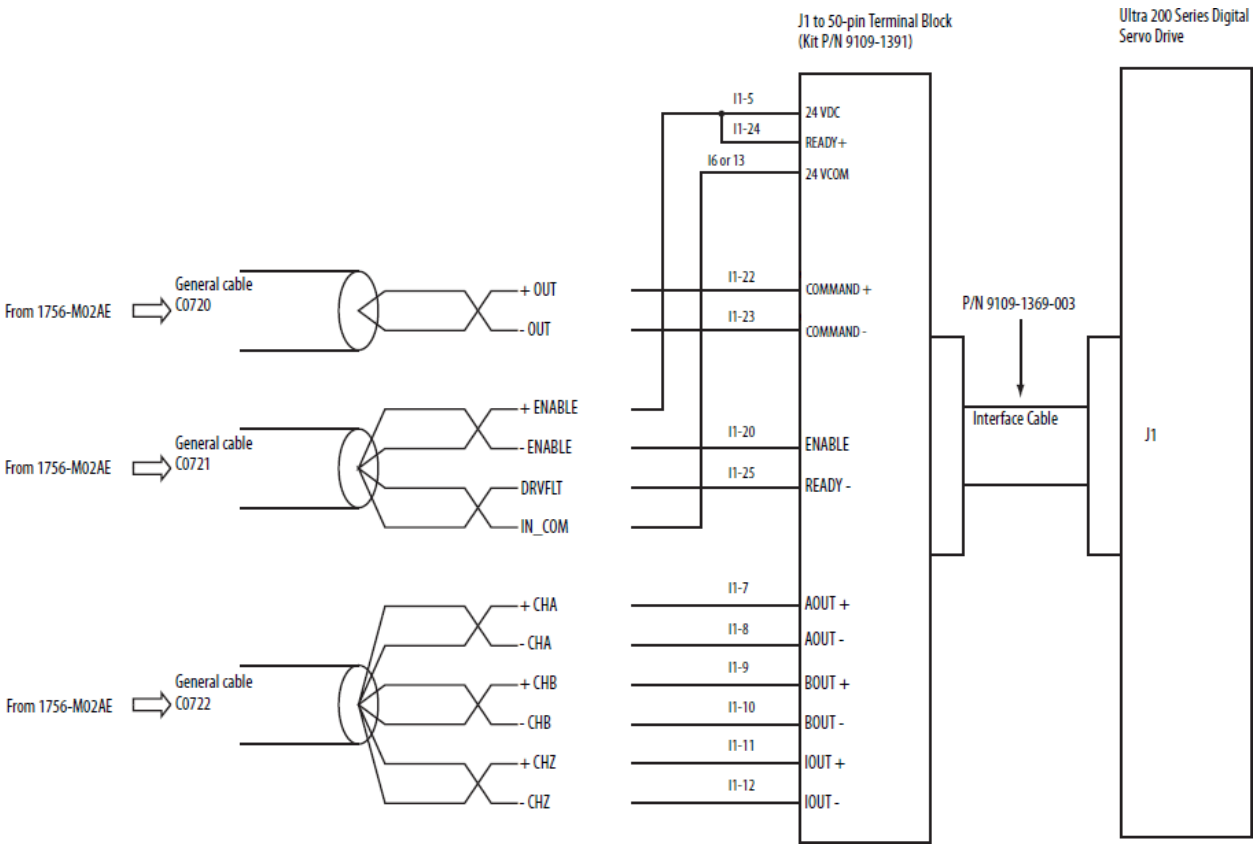


Notes

- This is an example of one way to wire the drive.
- See Ultra 100 Series Drive Installation Manual, publication 1398-5.2, for other configurations.

Ultra 200 Series Drive

The following image illustrates the Ultra 200 series drive.

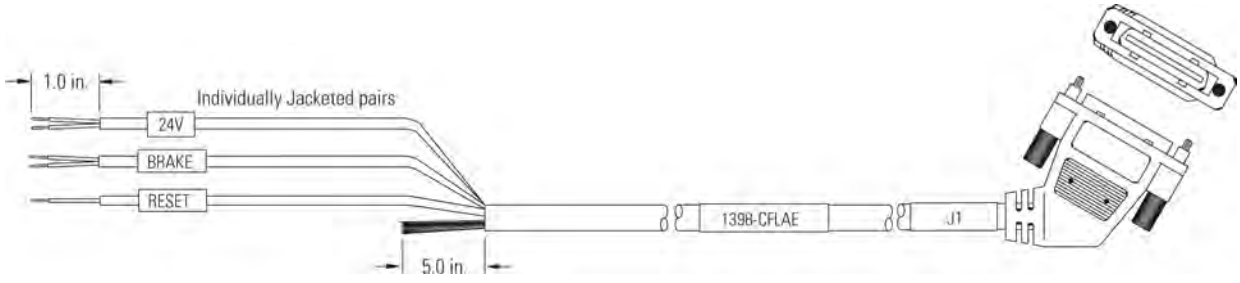


Notes

- This is an example of one way to wire the drive.
- See Ultra 200 Series Drive Installation Manual, publication number 1398-5.0, for other configurations.

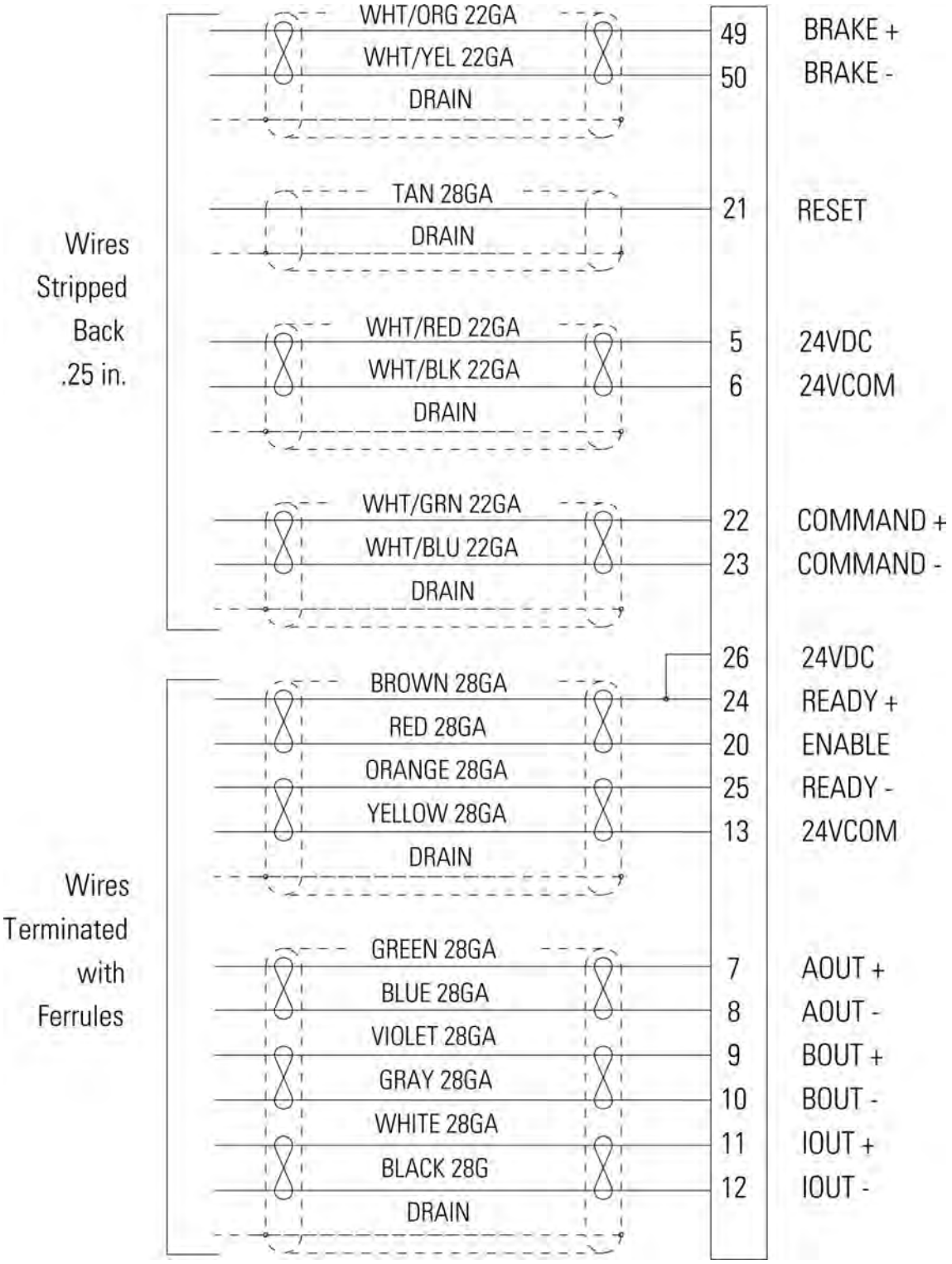
1398-CFLAExx cable

The following image illustrates the 1398-CFLAExx cable.



Pinouts for 1398-CFLAExx cable

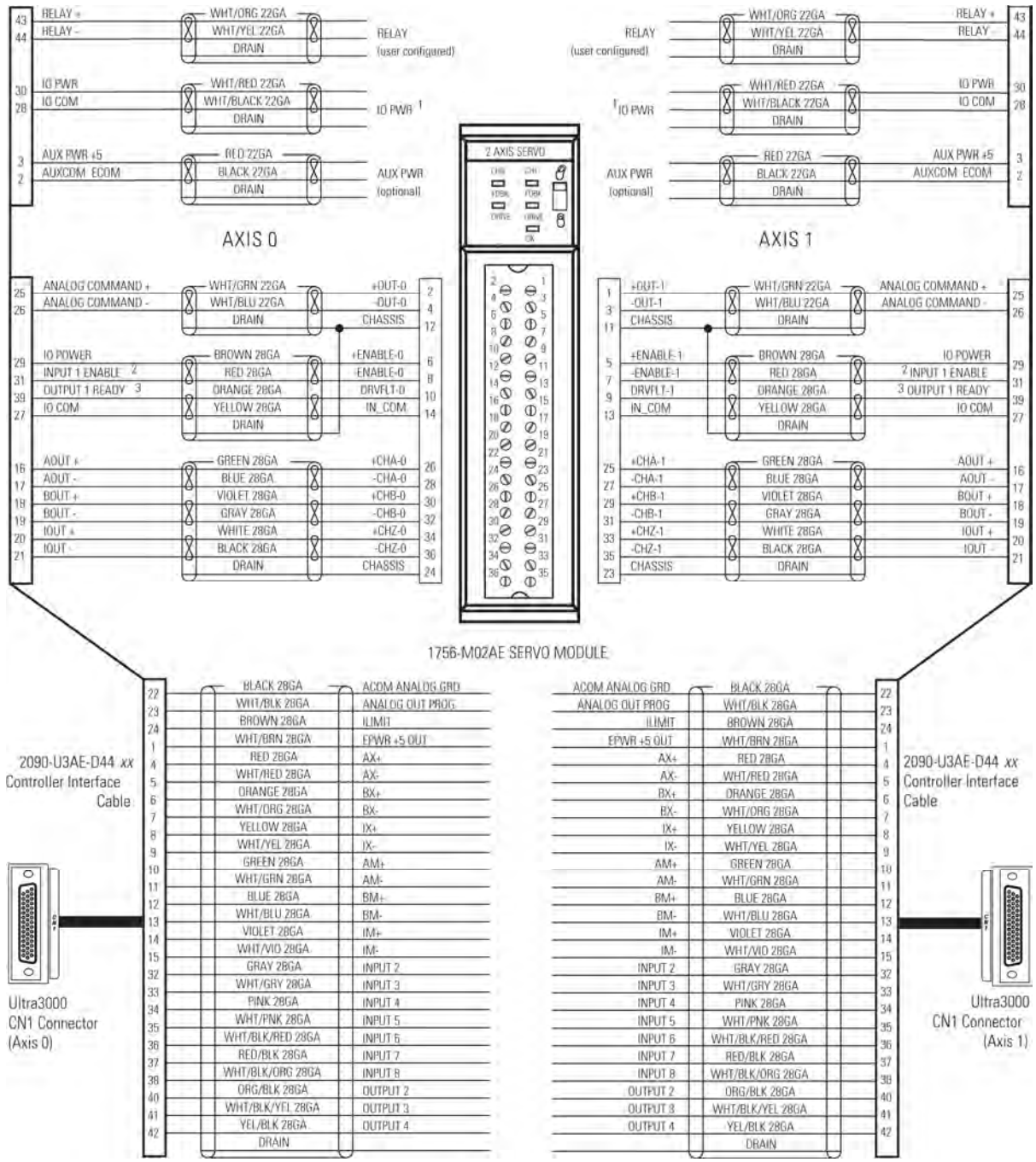
The following image illustrates the pinouts for the 1398-CFLAExx cable.



Ultra3000 Drive

Ultra3000 to 1756-M02AE interconnect diagram

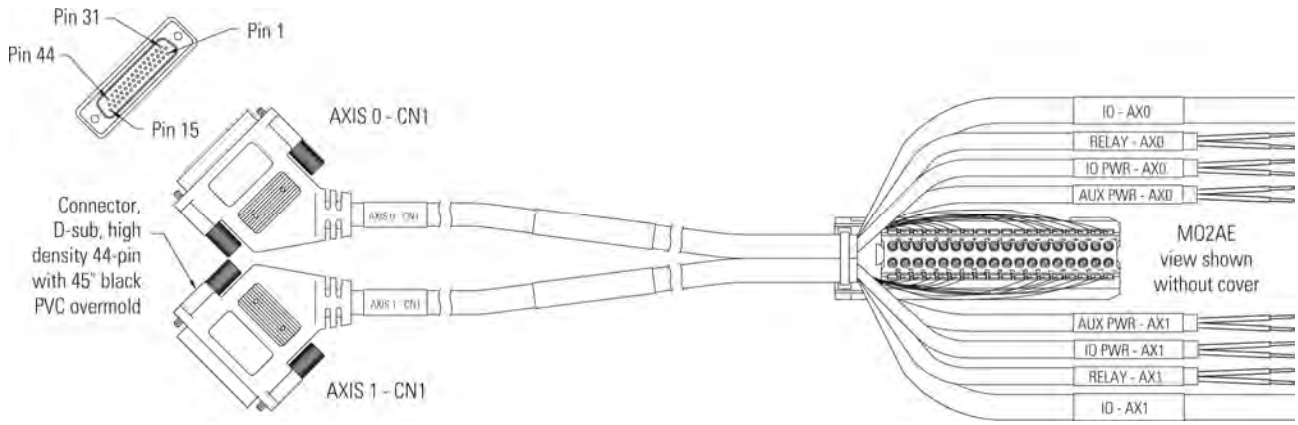
The following image illustrates interconnecting Ultra3000 to 1756-MO2AE.



For more information, see Ultra3000 Digital Servo Drives Installation Manual, publication 2098-IN003.

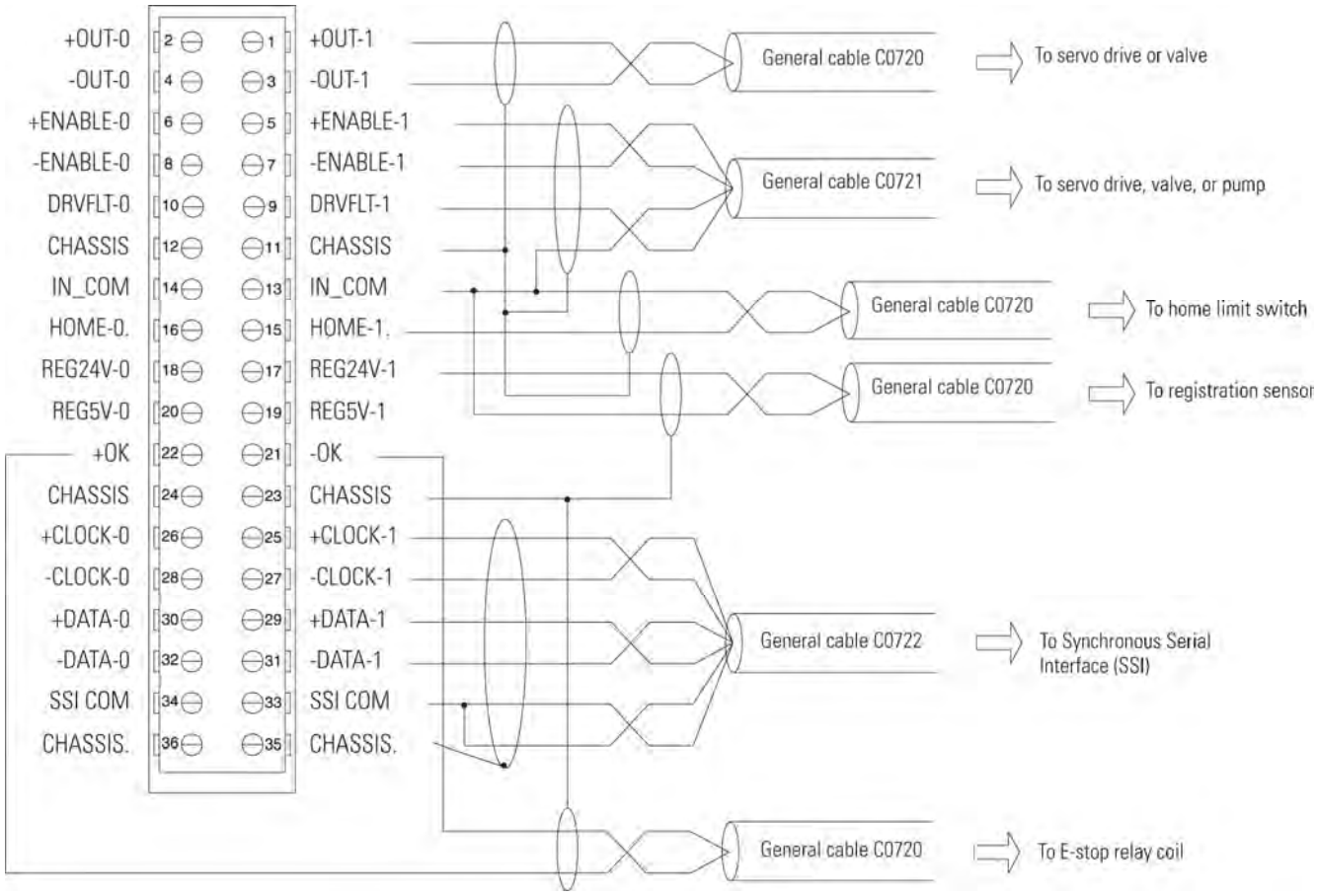
2090-U3AE-D44xx cable

The following image illustrates the 2090-U3AE-D44xx cable.



1756-M02AS module

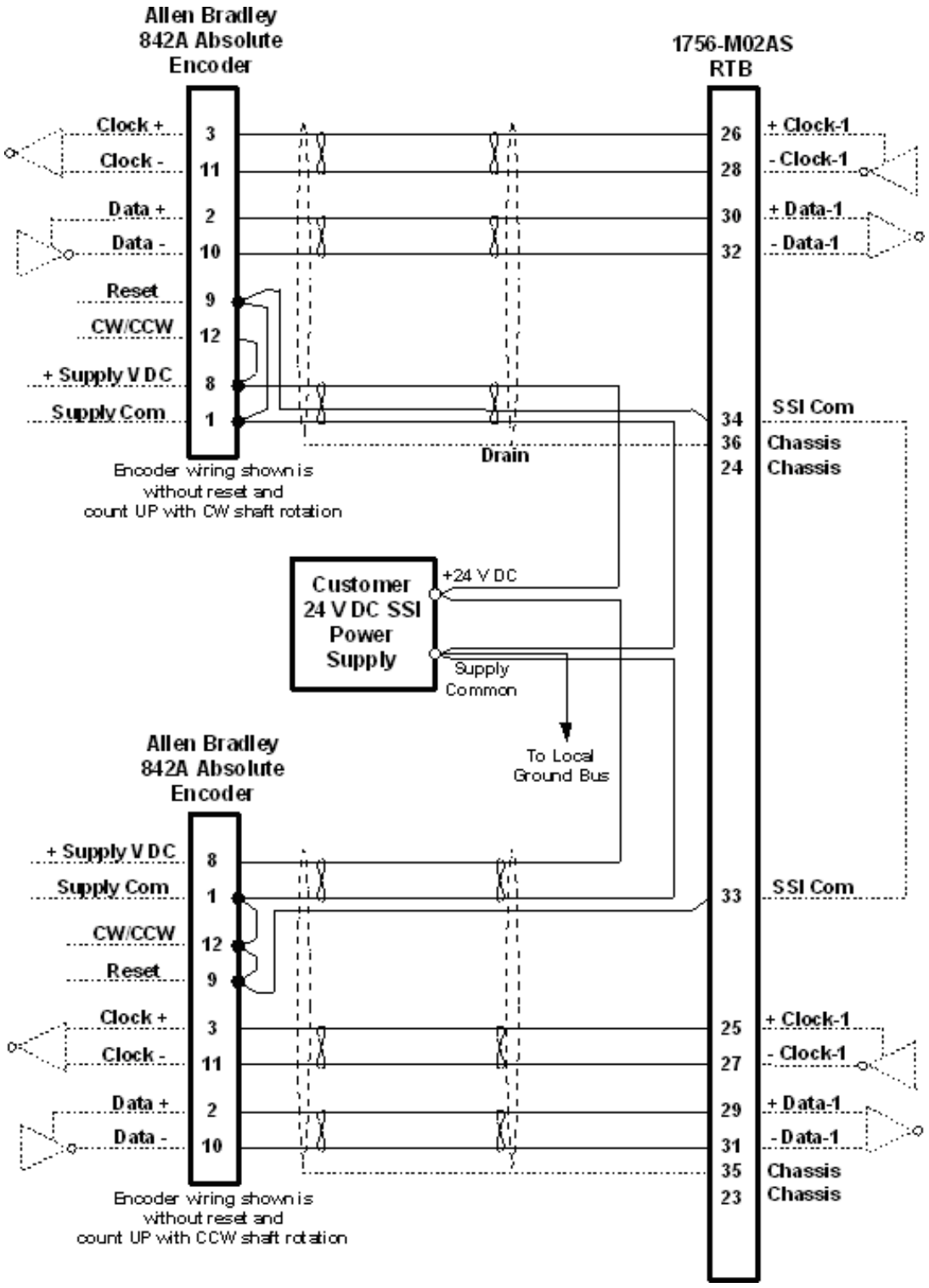
The following image illustrates the 1756-M02AS module.



This example shows the wiring for Axis 1. Wire Axis 0 the same way.

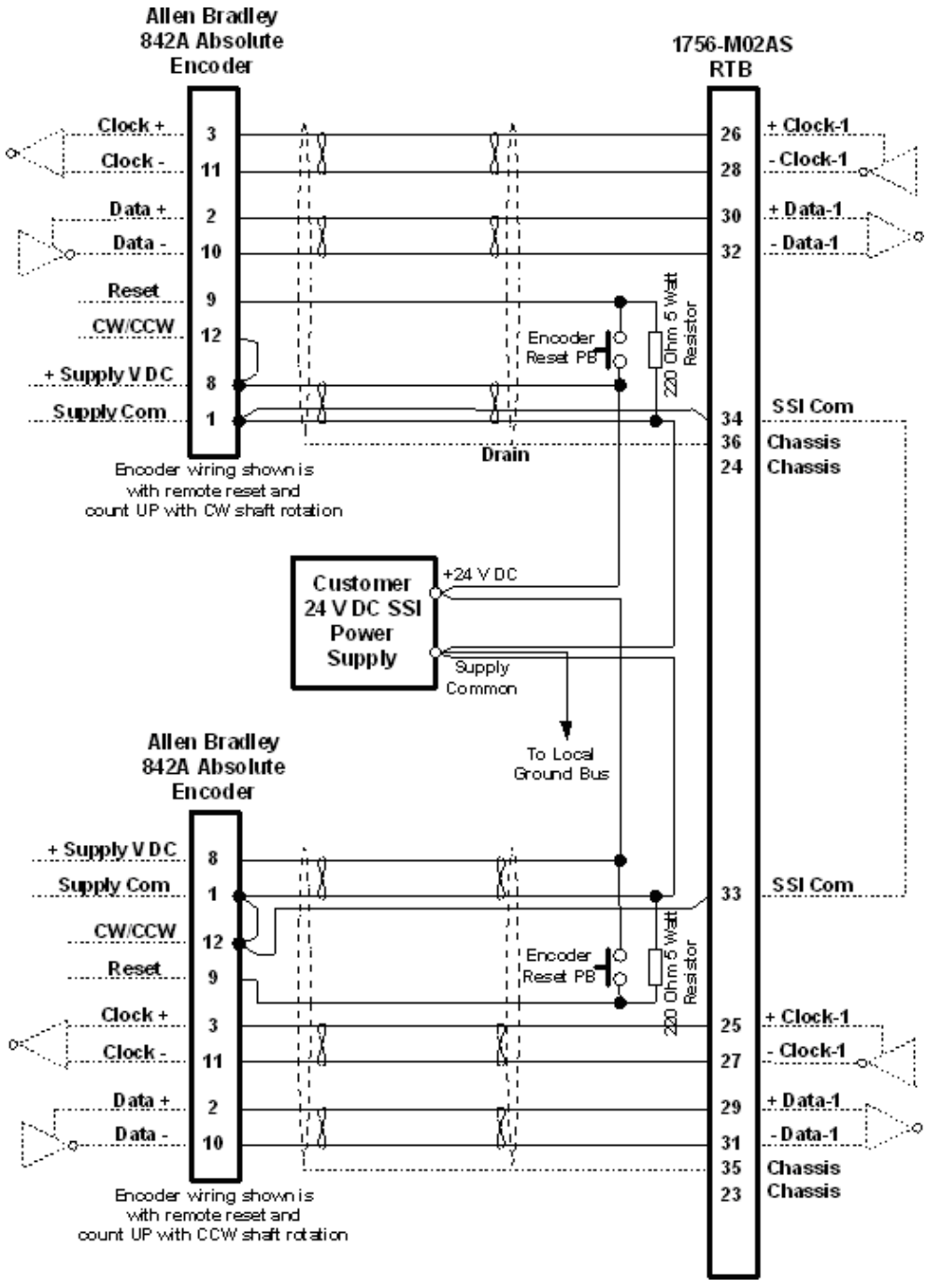
Wiring from AB 842A encoder without reset to 1756-M02AS RTB

The following image illustrates wiring from AB 842A encoder without reset to 1756-M02AS RTB.



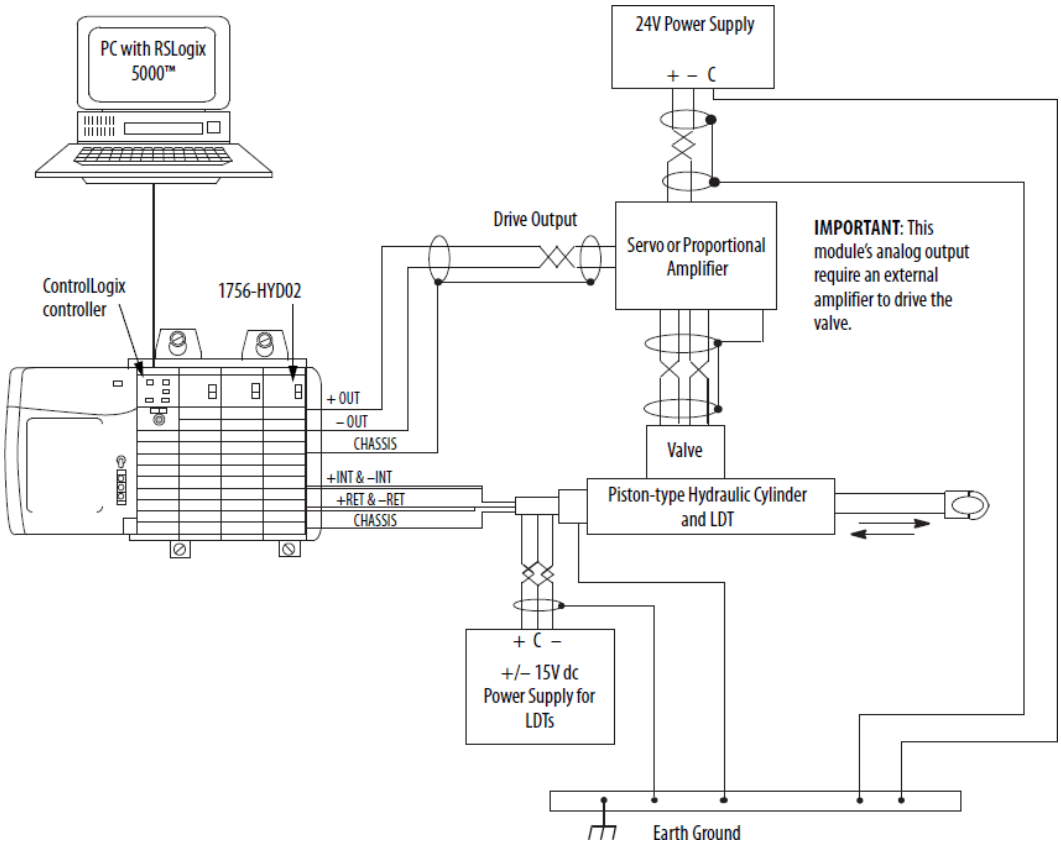
Wiring for AB 842A encoder with remote reset to 1756-M02AS RTB

The following image illustrates wiring for AB 842A encoder with remote reset to 1756-M02AS RTB.



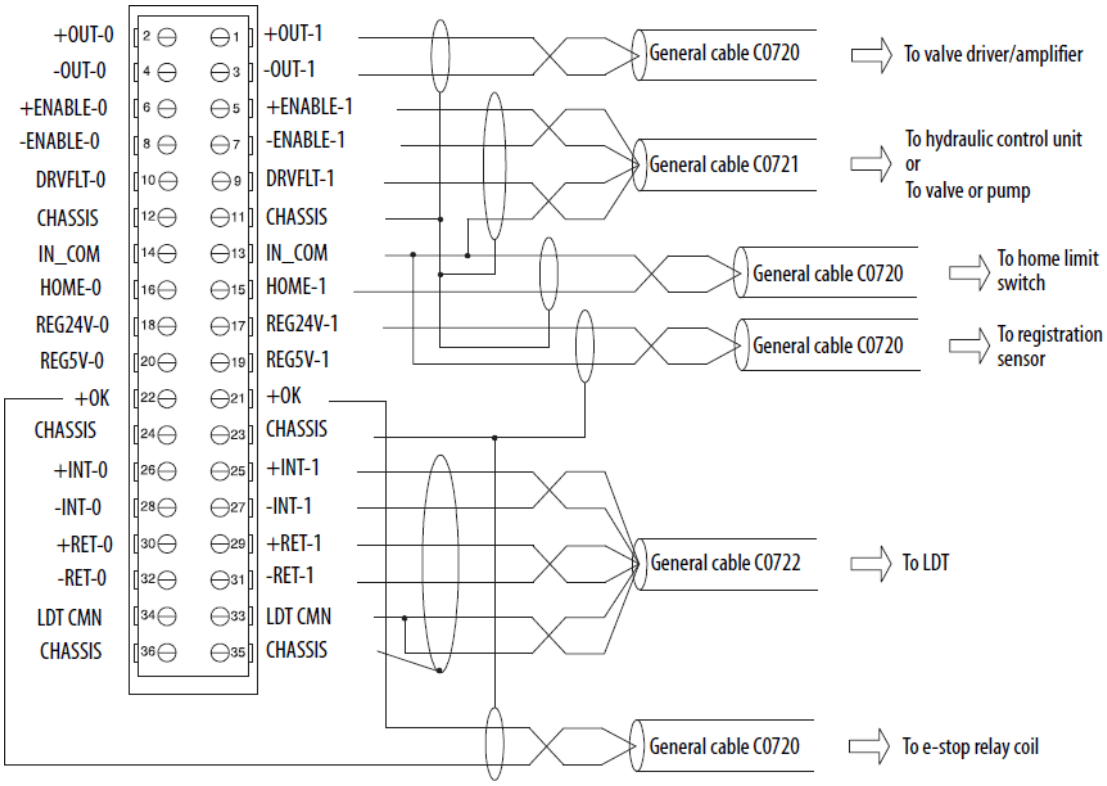
1756-HYD02 application example

The following image illustrates a 1-axis loop with a differential LDT input.



1756-HYD02 module

The following image illustrates the 1756-HYD02 module.



Notes

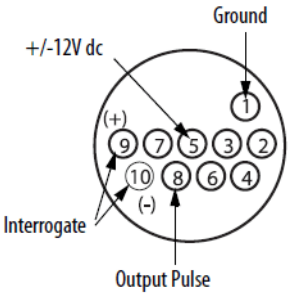
- This example shows the wiring for Axis 1. Wire Axis 0 the same way.
- Use transducers that use an external interrogation signal.
- Do not exceed the specified isolation voltage between power sources.

LDTs

These diagrams show the connections for Temposonic and Balluff LDTs.

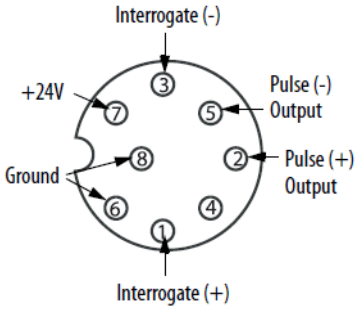
Important:	Other suppliers also have compatible LDTs. Before you connect an LDT to your module, make sure that it is the best LDT for your application.

Temposonics II, RPM or DPM

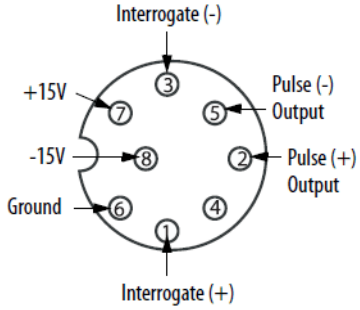


Balluff BTL type

24V Connections



+/- 15V Connections



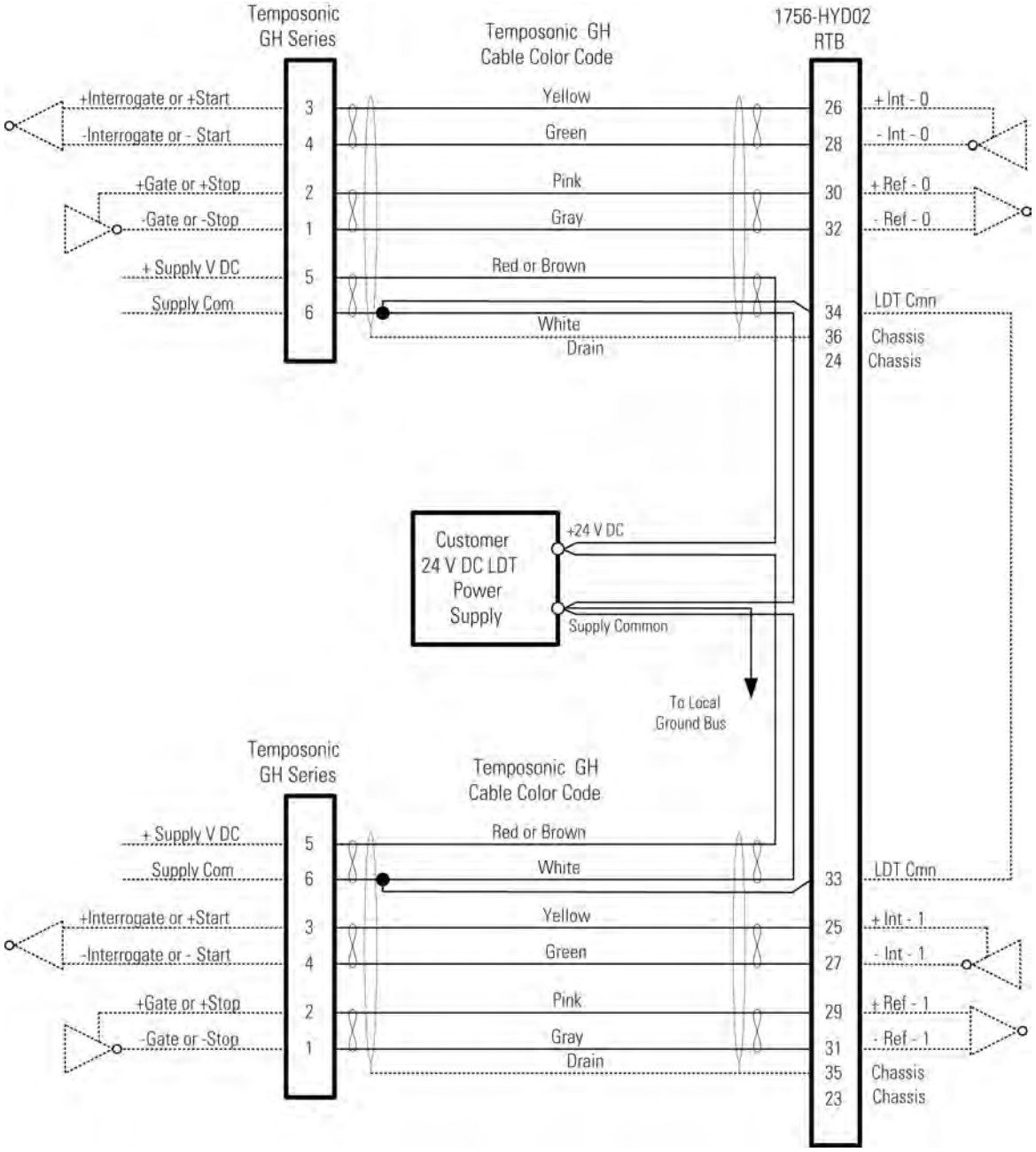
No shield connections on these examples

The following table lists the LDT connections for fabricating your won LDT cable.

Function(1)	**1756-HYD02 RTB Wiring (Numbers below represent terminal numbers)**		**Temposonics II(2) RPM or DPM**	**Balluff BTL type**	
	Channel 0	**Channel 1**		**24V dc**	**+/- 15V dc**
(+) Interrogate	26	25	9 - Yellow	1 - Yellow	1 - Yellow
(-) Interrogate	28	27	10 - Green	3 - Pink	3 - Pink
Power Supply	N/A		5 - Red (+/-12V)	7 - Brown (+24V)	7 - Brown (+15V) 8 - White (-15V)
Ground	34	33	1 - White	6 - Blue 8 - White	6 - Blue
Output Pulse	30 (+) 32 (-)	29 (+) 31 (-)	8 - Purple	2 - Gray (+) 5 - Green (-)	2 - Gray (+) 5 - Green (-)

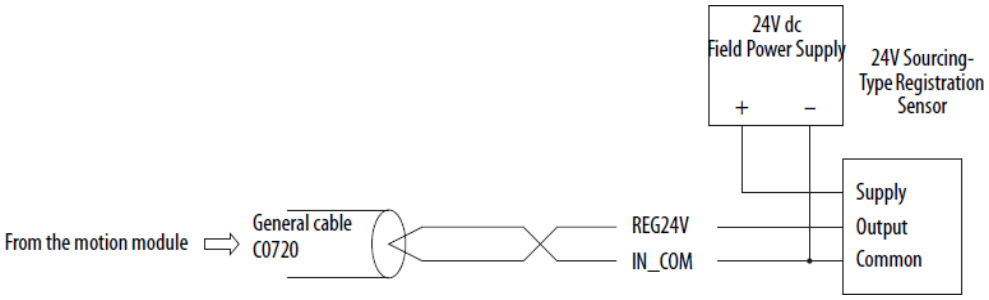
Temposonic GH feedback device

The following image illustrates the temposonic GH feedback device.



24V registration sensor

The following image illustrates the 24V registration sensor.

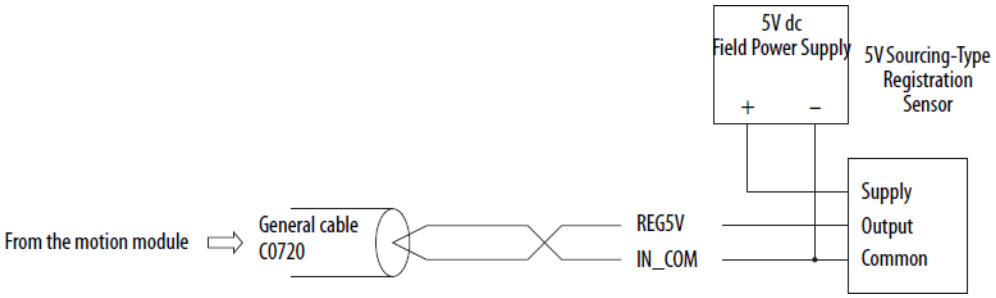


Notes

- Use sourcing-type registration sensors.
- Wire the inputs so that they get source current from the sensor.
- Do not use current sinking sensor configurations because the registration input common (IN_ COM) is shared with the other 24V servo module inputs.

5V registration sensor

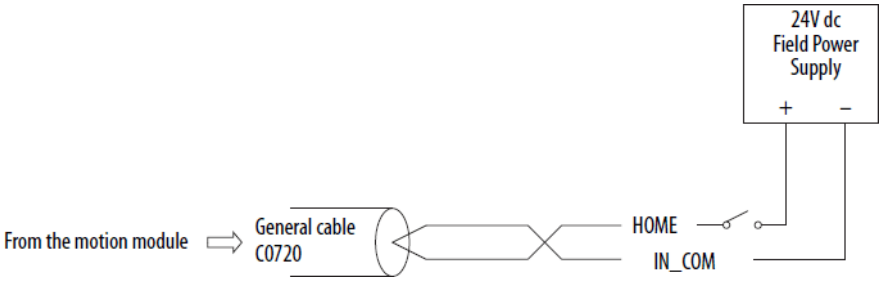
The following image illustrates the 5V registration sensor.



Notes

- Use sourcing-type registration sensors.
- Wire the inputs so that they get source current from the sensor.
- Do not use current sinking sensor configurations because the registration input common (IN_ COM) is shared with the other 24V servo module inputs.

Home limit switch input

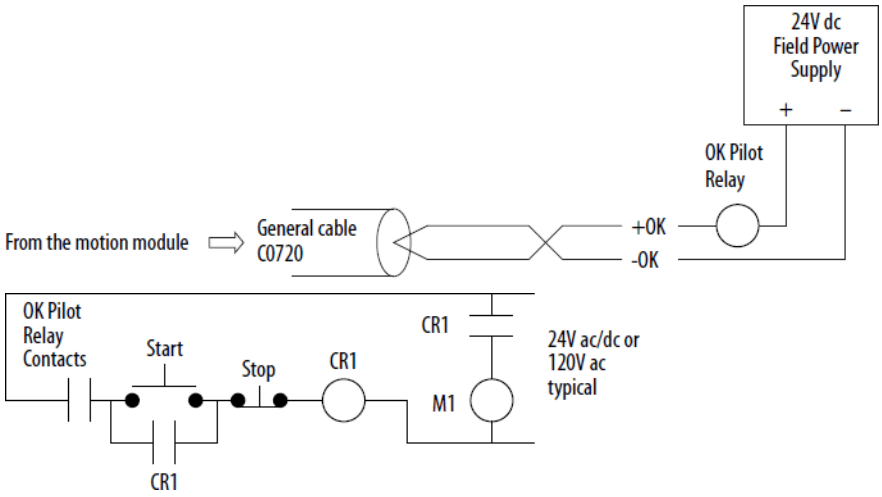


Notes

- The home limit switch inputs to the servo module are designed for 24V dc nominal operation.
- Wire these inputs for current sourcing operation.

OK contacts

The following image illustrates OK contacts.



Notes

- Use the OK relay contacts to connect to an E-stop string that controls power to the associated pumps or drives.
- The OK contacts are rated to drive an external 24V dc pilot relay (for example, Allen-Bradley 700-HA32Z24) whose contacts can be incorporated into the E-stop string.

Servo loop block diagrams

Introduction for Servo Loop Block Diagrams

This appendix shows the servo loop block diagrams for common motion configurations.

Interpreting the diagrams

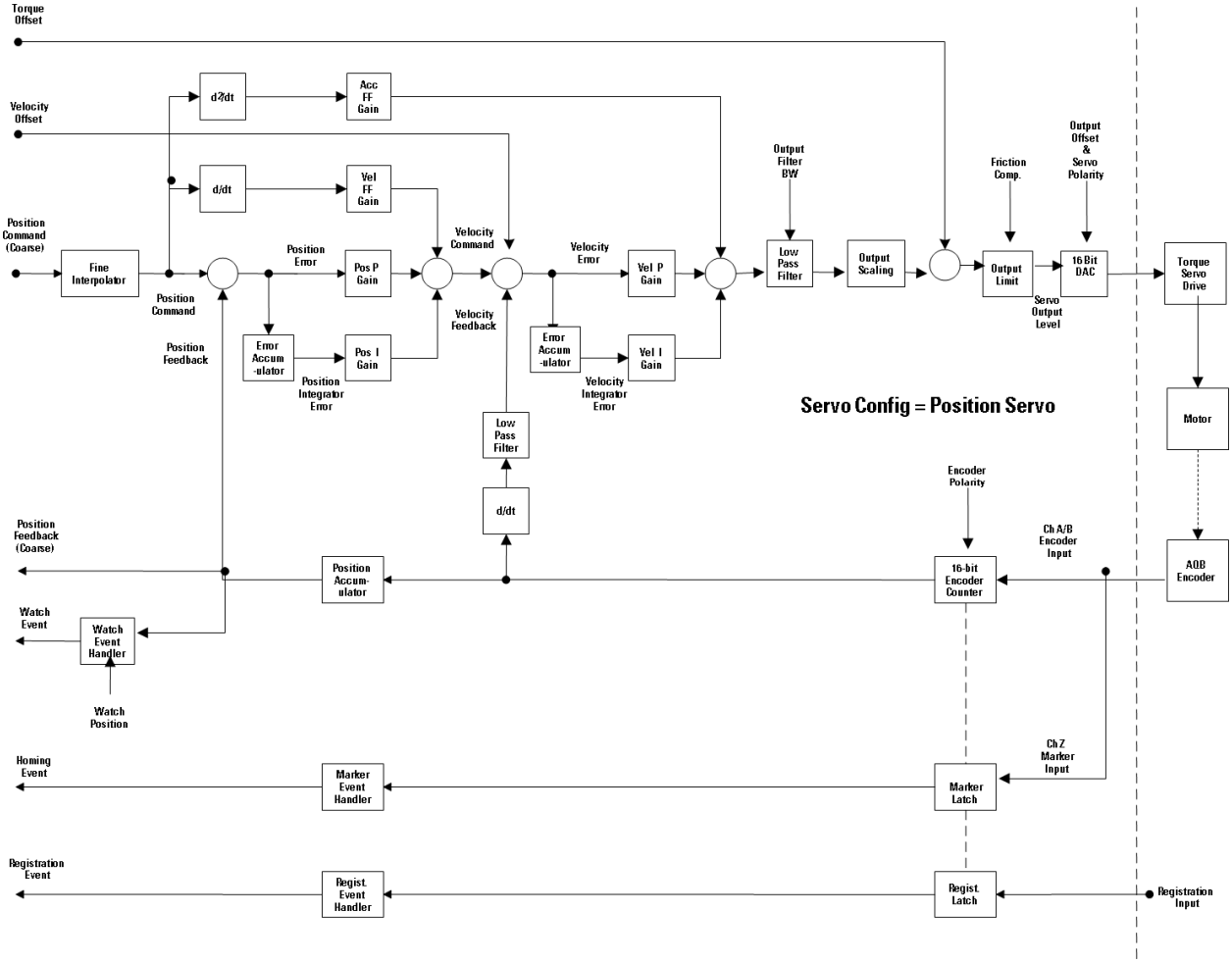
The diagrams use these labels for axes attributes.

Label	AXIS Attribute
Acc FF Gain	AccelerationFeedforwardGain
Friction Comp	FrictionCompensation
Output Filter BW	OutputFilterBandwidth
Output Limit	OutputLimit
Output Offset	OutputOffset
Output Scaling	OutputScaling
Pos I Gain	PositionIntegralGain
Pos P Gain	PositionProportionalGain
Position Error	PositionError
Position Integrator Error	PositionIntegratorError
Registration Position	RegistrationPosition
Servo Output Level	ServoOutputLevel
Vel FF Gain	VelocityFeedforwardGain
Vel I Gain	VelocityIntegralGain
Vel P Gain	VelocityProportionalGain
Velocity Command	VelocityCommand
Velocity Error	VelocityError
Velocity Feedback	VelocityFeedback
Velocity Integrator Error	VelocityIntegratorError
Watch Position	WatchPosition

AXIS_SERVO

Position servo with torque servo drive

The following image illustrates position servo with torque servo drive.

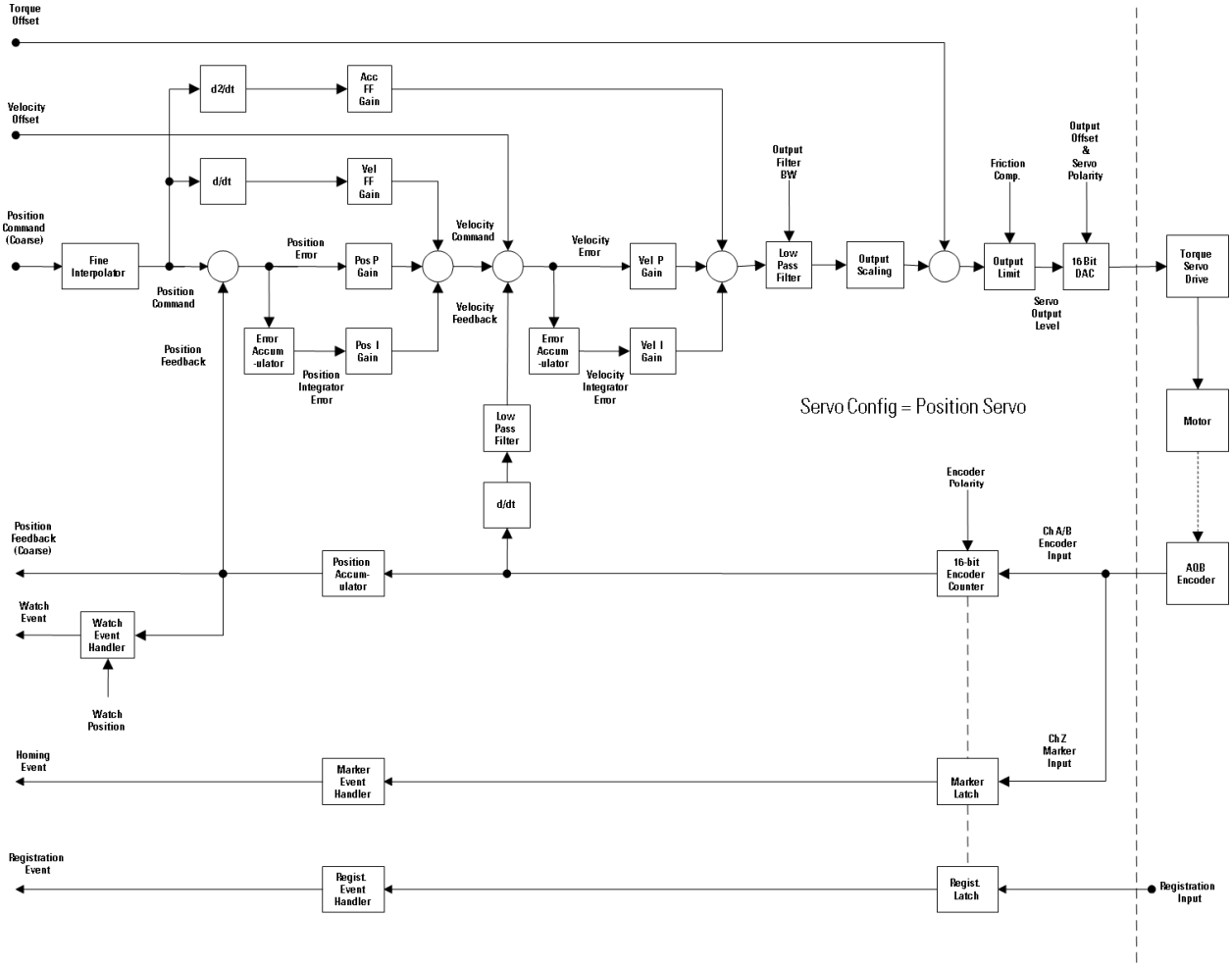


This configuration gives full position servo control using an external torque loop servo drive. Synchronous input data to the servo loop includes Position Command, Velocity Offset, and Torque Offset. The controller updates these values at the base update period of the motion group.

The Position Command value is derived directly from the output of the motion planner, whereas the Velocity Offset and Torque Offset values are derived from the current value of the corresponding attributes.

Position servo with velocity servo drive

The following image illustrates position servo with velocity servo drive.



This configuration provides full position servo control using an external velocity loop servo drive.

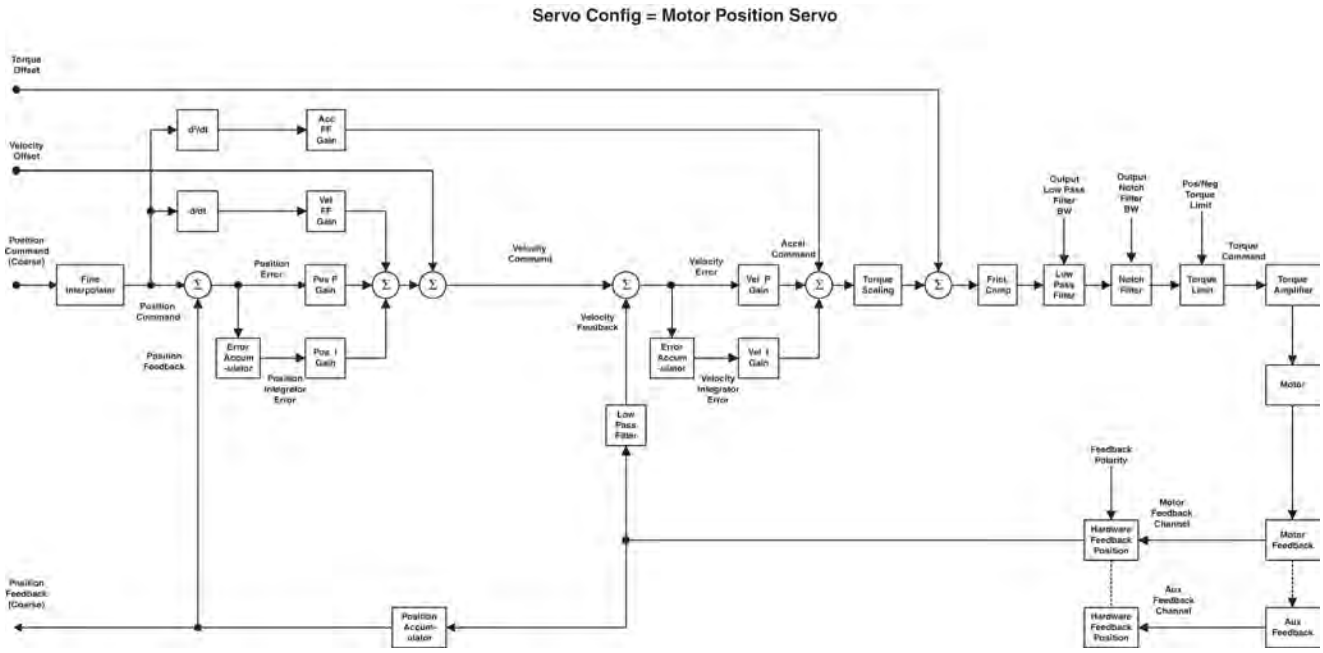
Note that in this configuration, the servo module does not close the velocity loop, but rather the drive does. Synchronous input data to the servo loop includes Position Command and Velocity Offset. (Torque Offset is ignored.) The controller updates these values at the base update period of the motion group.

The Position Command value is derived directly from the output of the motion planner, whereas the Velocity Offset value is derived from the current value of the corresponding attributes.

AXIS_SERVO_DRIVE

Motor Position Servo

The following image illustrates Motor Position Servo configuration.



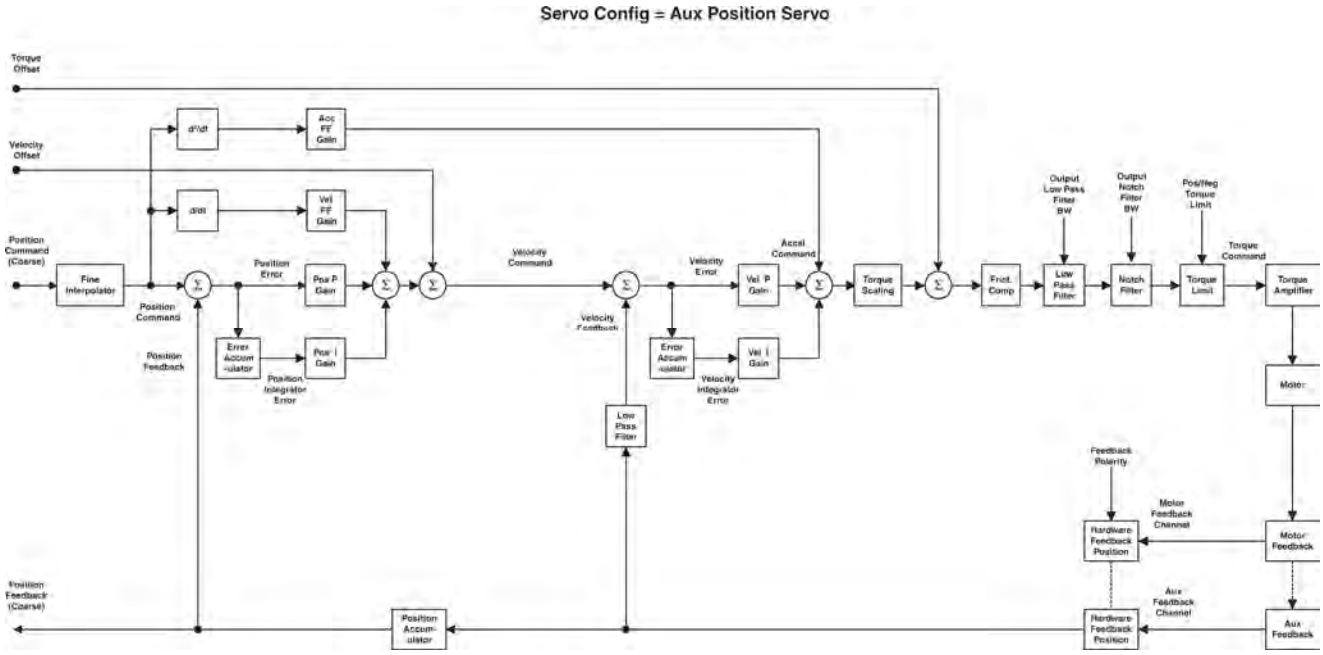
The Motor Position Servo configuration provides full position servo control using only the motor mounted feedback device to provide position and velocity feedback. This servo configuration is a good choice in applications where smoothness and stability are more important that positioning accuracy. Positioning accuracy is limited due to the fact that the controller has no way of compensating for non-linearity in the mechanics external to the motor.

Note that the motor mounted feedback device also provides motor position information necessary for commutation. Synchronous input data to the servo loop includes Position Command, Velocity Offset, and Torque Offset. These values are updated at the base update rate of the associated motion group.

The Position Command value is derived directly from the output of the motion planner, while the Velocity Offset and Torque Offset values are derived from the current value of the corresponding attributes. These offset attributes may be changed programmatically via SSV instructions or direct Tag access, which, when used in conjunction with future Function Block programs, provides custom ‘outer’ control loop capability.

Auxiliary Position Servo

The following image illustrates Auxiliary Position Servo configuration.



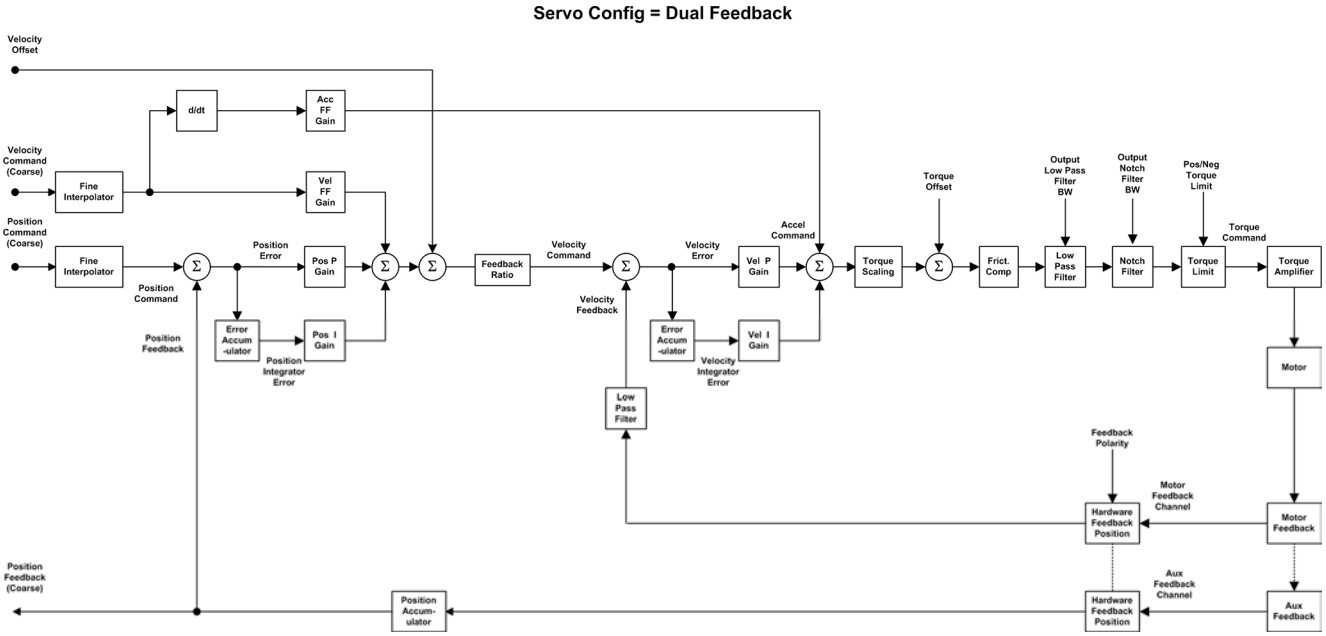
The Auxiliary Position Servo configuration provides full position servo control using an auxiliary (that is, external to the motor) feedback device to provide position and velocity feedback. This servo configuration is a good choice in applications where positioning accuracy is important. The smoothness and stability may be limited, however, due to the mechanical non-linearities external to the motor.

Note that the motor mounted feedback device is still required to provide motor position information necessary for commutation. Synchronous input data to the servo loop includes Position Command, Velocity Offset, and Torque Offset. These values are updated at the base update rate of the associated motion group.

The Position Command value is derived directly from the output of the motion planner, while the Velocity Offset and Torque Offset values are derived from the current value of the corresponding attributes. These offset attributes may be changed programmatically via SSV instructions or direct Tag access, which, when used in conjunction with future Function Block programs, provides custom 'outer' control loop capability.

Dual Position Servo

The following image illustrates Dual Position Servo configuration.



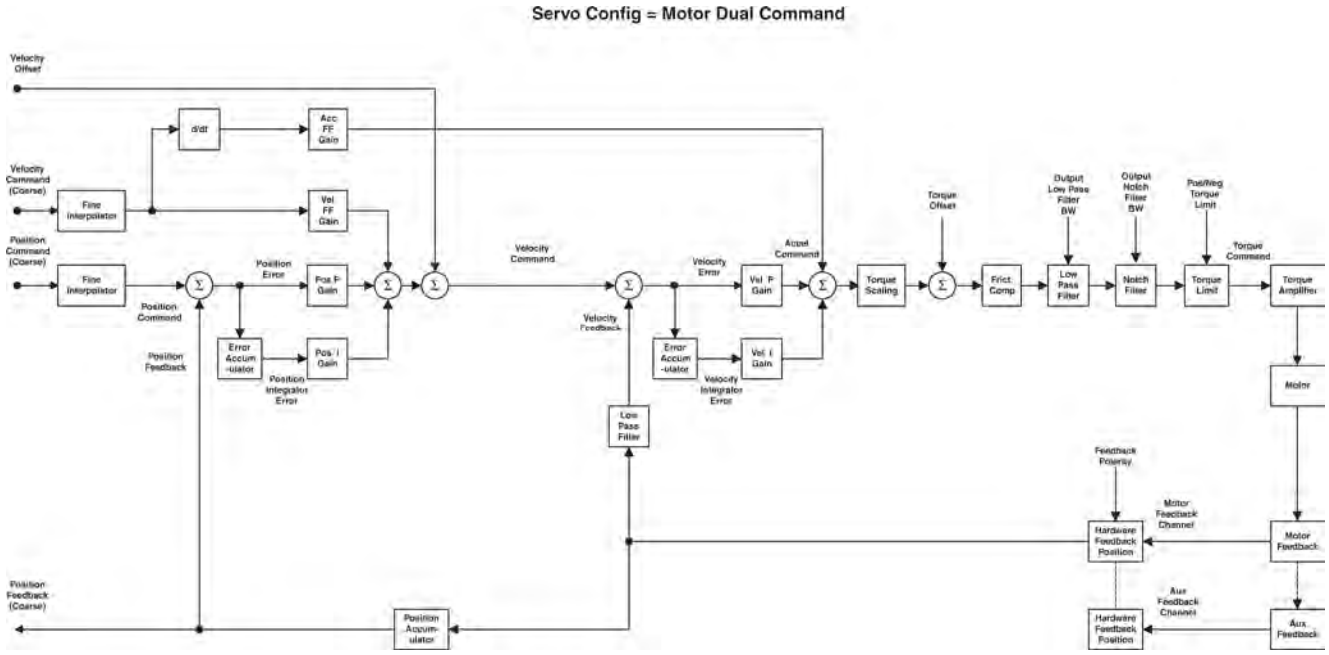
This configuration provides full position servo control using the auxiliary feedback device for position feedback and the motor mounted feedback device to provide velocity feedback. This servo configuration combines the advantages of accurate positioning associated with the auxiliary position servo with the smoothness and stability of the motor position servo configuration.

Note that the motor mounted feedback device also provides motor position information necessary for commutation. Synchronous input data to the servo loop includes Position Command, Velocity Offset, and Torque Offset. These values are updated at the base update rate of the associated motion group.

The Position Command value is derived directly from the output of the motion planner, while the Velocity Offset and Torque Offset values are derived from the current value of the corresponding attributes. These offset attributes may be changed programmatically via SSV instructions or direct Tag access, which, when used in conjunction with future Function Block programs, provides custom 'outer' control loop capability.

Motor Dual Command Servo

The following image illustrates Motor Dual Command Servo configuration.



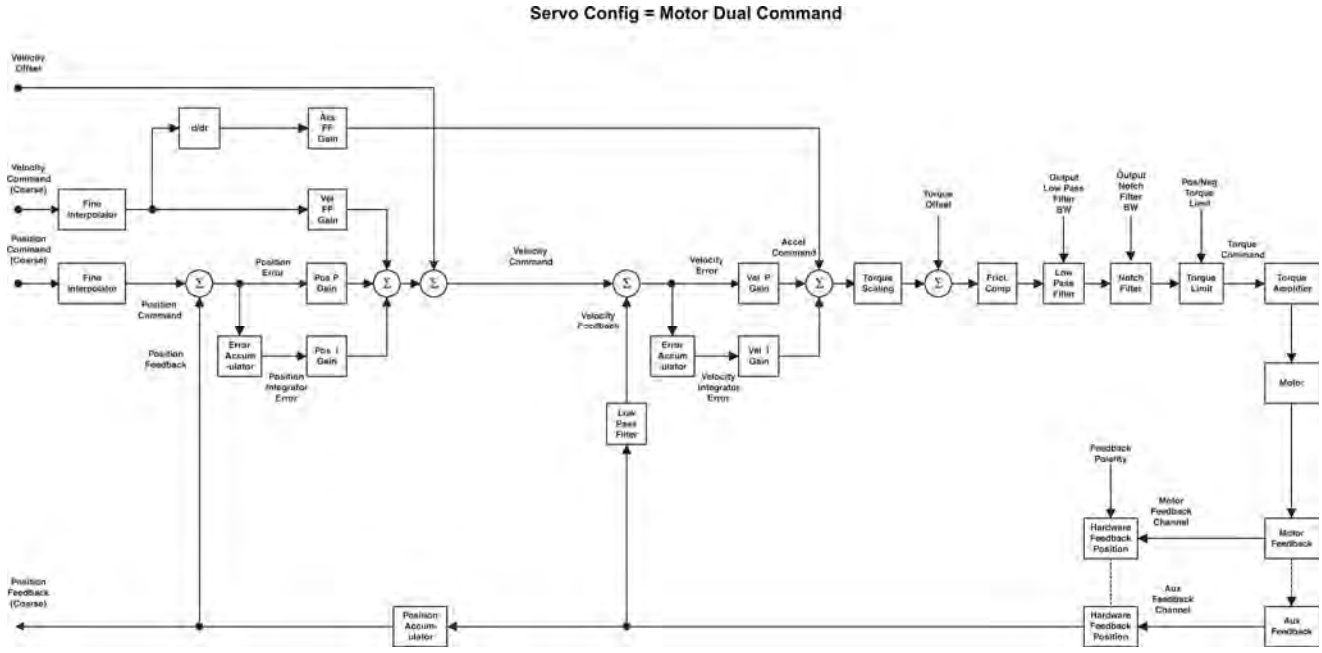
The Motor Dual Command Servo configuration provides full position servo control using only the motor mounted feedback device to provide position and velocity feedback. Unlike the Motor Position Servo configuration, however, command position and command velocity are applied to the loop to provide smoother feedforward behavior. This servo configuration is a good choice in applications where smoothness and stability are important. Positioning accuracy is limited due to the fact that the controller has no way of compensating for non-linearities in the mechanics external to the motor.

Note that the motor mounted feedback device also provides motor position information necessary for commutation. Synchronous input data to the servo loop includes Position Command, Velocity Command, and Velocity Offset. These values are updated at the base update rate of the associated motion group.

The Position and Velocity Command values are derived directly from the output of the motion planner, while the Velocity Offset value is derived from the current value of the corresponding attributes. The velocity offset attribute may be changed programmatically via SSV instructions or direct Tag access, which, when used in conjunction with future Function Block programs, provides custom 'outer' control loop capability.

Auxiliary Dual Command Servo

The following image illustrates Auxiliary Dual Command Servo configuration.



The Auxiliary Dual Command Servo configuration provides full position servo control using only the auxiliary mounted feedback device to provide position and velocity feedback. Unlike the Auxiliary Position Servo configuration, however, command position and command velocity are applied to the loop to provide smoother feedforward behavior. This servo configuration is a good choice in applications where positioning accuracy and good feedforward performance is important. The smoothness and stability may be limited, however, due to the mechanical non-linearities external to the motor.

Note, that the motor mounted feedback device is still required to provide motor position information necessary for commutation. Synchronous input data to the servo loop includes Position Command, Velocity Command, and Velocity Offset. These values are updated at the base update rate of the associated motion group.

The Position and Velocity Command values are derived directly from the output of the motion planner, while the Velocity Offset value is derived from the current value of the corresponding attributes. The velocity offset attribute may be changed programmatically via SSV instructions or direct Tag access, which, when used in conjunction with future Function Block programs, provides custom 'outer' control loop capability.

Dual Command Feedback Servo

The Motor Dual Command Feedback Servo configuration provides full position servo control using the auxiliary feedback device for position feedback and the motor mounted feedback device to provide velocity feedback. Unlike the Dual Feedback Servo configuration, however, command position and command velocity are also applied to the loop to provide smoother feedforward behavior. This servo configuration is a good choice in applications where smoothness, stability, and positioning accuracy are important.

Note, that the motor mounted feedback device is still required to provide motor position information necessary for commutation. Synchronous input data to the servo loop includes Position Command, Velocity Command, and Velocity Offset. These values are updated at the base update rate of the associated motion group.

The Position and Velocity Command values are derived directly from the output of the motion planner, while the Velocity Offset value is derived from the current value of the corresponding attributes. The velocity offset attribute may be changed programmatically via SSV instructions or direct Tag access, which, when used in conjunction with future Function Block programs, provides custom 'outer' control loop capability.

Velocity Servo

The Velocity Servo configuration provides velocity servo control using the motor mounted feedback device. Synchronous input data to the servo loop includes Velocity Command, Velocity Offset, and Torque Offset. These values are updated at the base update rate of the associated motion group. The Velocity Command value is derived directly from the output of the motion planner, while the Velocity Offset and Torque Offset values are derived from the current value of the corresponding attributes. These offset attributes may be changed programmatically via SSV instructions or direct Tag access which, when used in conjunction with future Function Block programs, provides custom 'outer' control loop capability.

Torque Servo

The Torque Servo configuration provides torque servo control using only the motor mounted feedback device for commutation. Synchronous input data to the

servo loop includes only the Torque Offset. These values are updated at the base update rate of the associated motion group.

The Torque Offset value is derived from the current value of the corresponding attribute. This offset attribute may be changed programmatically via SSV instructions or direct Tag access, which, when used in conjunction with future Function Block programs, provides custom 'outer' control loop capability.

Drive Gains

Rockwell Automation servo drives use Nested Digital Servo Control Loop, such as shown in the block diagrams, consisting typically of a position loop with proportional, integral, and feed-forward gains around a digitally synthesized inner velocity loop, again with proportional and integral gains for each axis.

These gains provide software control over the servo dynamics, and allow the servo system to be completely stabilized. Unlike analog servo controllers, these digitally set gains do not drift. Furthermore, once these gains are set for a particular system, another SERCOS module programmed with these gain values operates identically to the original one.

1

1398-CFLAExx
Cable Diagram 317
Pinouts 318
1756-M02AE servo module
Block diagrams
Torque servo drive 332
Velocity servo drive 333
Features 14
Loop and interconnect diagrams 331
1756-M03SE
set up 42, 52, 56
1756-M08SE
set up 42, 52, 56
1756-M16SE
set up 42, 52, 56

A

Analog Drives
set up 52
Analog interface drive
add to controller 51
Attributes
replicated 208
axis
add to controller 40, 42
check wiring 65
get status 42, 62
inhibit 71
set up 42
Axis Properties
Aux Feedback Tab (AXIS_SERVO_DRIVE)
Cycles 145
Feedback Ratio 145
Feedback Type 145
Interpolation Factor 145
Per 145
Conversion Tab
Conversion Constant 146
Position Unwind 146
Positioning Mode 146
Drive/Motor Tab - (AXIS_SERVO_DRIVE)
Amplifier Catalog Number 140
Calculate button 140
Calculate Parameters 140
Conversion Constant 140
Drive Resolution 140
Per 140
Position Range 140
Position Unit Scaling 140
Position Unit Unwind 140
Change Catalog Button 140
Catalog Number 140
Filters 140
Family 140
Feedback Type 140
Voltage 140
Loop Configuration 140
Drive/Motor Tab (AXIS_SERVO_DRIVE)
(Motor) Catalog Number 140
Dynamics Tab
Calculate
Maximum Acceleration Jerk
Maximum Deceleration Jerk 163
Manual Tune 163
Maximum Acceleration 161
Maximum Acceleration Jerk 161
Maximum Deceleration 161
Maximum Deceleration Jerk 162
Maximum Velocity 160
Fault Actions Tab - AXIS_SERVO
Drive Fault 197
Feedback Loss 197

Feedback Noise 197

Position Error 197

Soft Overtravel 198

Fault Actions Tab - AXIS_SERVO_DRIVE

Drive Thermal 199

Feedback 200

Feedback Noise 200

Hard Overtravel 200

Motor Thermal 200

Position Error 200

Set Custom Stop Action 201

Soft Overtravel 200

Feedback Tab - AXIS_SERVO

Feedback Type 137

A Quadrature B Encoder Interface (AQB 137
Linear Displacement Transducer (LDT) 137
Absolute Feedback Offset 137
Calculated Values 137
Calculate Button 137
Conversion Constant 137
Minimum Servo Update Period 137
Calibration Constant 137
Enable Absolute Feedback 137
LDT Type 137
Length 137
Recirculations 137
Scaling 137
Synchronous Serial Interface (SSI)
Absolute Feedback Offset 137
Clock Frequency 137
Code Type 137
Data Length 137
Enable Absolute Feedback 137
Synchronous Serial Interface(SSI) 137

Gains Tab - AXIS_SERVO

Differential 166

Integral (Position) Gain 165

Integrator Hold 168

Manual Tune 168

Proportional (Position) Gain 165

Proportional (Velocity) Gain 166

Gains Tab - AXIS_SERVO_DRIVE

Integral (Position) Gain 171

Integral (Velocity) Gain 166, 172

Integrator Hold 172

Manual Tune 173

Proportional (Position) Gain 170

Proportional (Velocity) Gain 166, 171

Set Custom Gains 173

Velocity Feedforward 167, 170

Homing Tab - SERVO_AXIS and SERVO_AXIS_DRIVE

Direction 147, 150

Limit Switch 147, 150

Mode 147

Offset 147, 150

Position 147, 150

Return Speed 147, 150

Sequence 147

Speed 147, 150

Hookup Tab - AXIS_SERVO

Feedback Polarity 153

Output Polarity 153

Test Feedback 153

Test Increment 153

Test Marker 153

Test Output & Feedback 153

Hookup Tab Overview - AXIS_SERVO_DRIVE
Drive Polarity 155

Test Feedback 156

Test Increment 155

Test Marker 156

Test Output & Feedback 156

Limits Tab - AXIS_SERVO
Manual Tune 183

Maximum Negative 181

Maximum Positive 181

Output Limit 182

Position Error Tolerance 181

Soft Travel Limits 181

Limits Tab - AXIS_SERVO_DRIVE
Continuous Torque/Force Limit 185

Hard Travel Limits 184

Manual Tune 186

Maximum Negative 184

Maximum Positive 184

Peak Torque/Force Limit 185

Position Error Tolerance 185

Position Lock Tolerance 185

Set Custom Limits 186

Soft Travel Limits 184

Motor/Feedback Tab (AXIS_SERVO_DRIVE) 145
(Motor) Cycles 145

(Motor) Feedback Type 145

(Motor) Interpolation Factor 145

Per 145

Offset Tab - AXIS_SERVO
Backlash Compensation 190

Reversal Offset 190
Stabilization Window 191
Friction/Deadband Compensation 190

Friction Compensation 190
Friction Compensation Window 190
Manual Tune 192

Output Offset 191

Torque Offset 191

Velocity Offset 191

Offset Tab - AXIS_SERVO_DRIVE
Backlash Compensation 194

Reversal Offset 194
Stabilization Window 195
Friction Compensation 193

Friction Compensation Window 194

Manual Tune 195

Torque Offset 195

Velocity Offset 195

Output Tab - SERVO_AXIS
Enable Low-pass Output Filter 176

Low-pass Output Filter Bandwidth 176

Manual Tune 177

Torque Scaling 175

Velocity Scaling 175

Output Tab Overview - AXIS_SERVO_DRIVE
Enable Low-pass Output Filter 179

Enable Notch Filter 179

Load Inertia Ratio 178

Low-pass Output Filter Bandwidth 179

Manual Tune 180

Motor Inertia 178

Notch Filter 179

Torque Scaling 178

Servo Tab - AXIS_SERVO

Direct Drive Ramp Rate 136

Drive Fault Input 136

Enable Direct Drive Ramp Control 136

Enable Drive Fault Input 136

External Drive Configuration 136

Hydraulic 136
Torque 136
Velocity 136

Loop Configuration 136

Real Time Axis Information

Attribute 1/Attribute 2 136

Tag Tab

Data Type 203

Description 203

Name 203

Scope 204

Style 204

Tag Type 203

Tune Tab - AXIS_SERVO, AXIS_SERVO_DRIVE

Damping Factor 158

Direction 157

Speed 157

Start Tuning 159

Torque (AXIS_SERVO) 157

Torque/Force (AXIS_SERVO_DRIVE) 157

Travel Limit 157

Tune 158

B

Block diagrams for a 1756-M02AE module 331
With a torque servo drive 332
With a velocity servo drive 333

C

CIP Sync 28
publication 28
configure
configure 42, 52, 56
coordinate system
overview 42, 62
CST, PTP 28

D

data types
MOTION_GROUP 132
diagrams
Block 331
wiring 315
Direct Commands
Supported Commands
Motion State 78, 101

drive
add Analog interface drive 51
add SERCOS interface drive 33
check wiring 65

E

Editing Axis Properties
General Tab - AXIS_GENERIC
Axis Configuration 133

Channel 133

Ellipsis (?) button 133
Module 133
Motion Group 133
General Tab - AXIS_SERVO_DRIVE
Assigned Motion Group 130
Axis Configuration 130
Module 130
Module Type 130
Node 130
Node with a Kinetix 6000 Drive 131
General Tab - SERVO_AXIS
Axis Configuration 129
Channel 129
Module 129
Motion Planner Tab
Enable Master Position Filter Checkbox 134
Master Delay Compensation Checkbox 134
Master Position Filter Bandwidth 134
Output Cam Execution Targets 134
Program Stop Action 134
Units Tab
Average Velocity Timebase 136
Position Units 136

Encoder 209
Noise 209

G

General Tab - AXIS_VIRTUAL
Assigned Motion Group 132
Grandmaster 28

H

home limit switch input wire 329
Homing 117
hookup tests
run 65
Hydraulic Drives
set up 56

I

inhibit
axis 71

L

Linear displacement transducer (LDT)
Connecting the LDT to the 1756-HYD02 module 325, 326

M

Motion Apply Axis Tuning 78, 101
Motion Apply Hookup Diagnostic 78, 101
Motion Arm Output Cam 78, 101
Motion Arm Registration 78, 101
Motion Arm Watch Position 78, 101
Motion Attributes
Axis Event Bit Attributes 209
Axis Fault Bit Attributes 209
Axis Status Bit Attributes 209
Commissioning Configuration Attributes
Damping Factor 209
Drive Model Time Constant 209
Position Servo Bandwidth 209
Test Increment 209
Tuning Configuration Bits 209
Bi-directional Tuning 209
Tune Acceleration Feedforward 209
Tune Friction Compensation 209
Tune Output Low-Pass Filter 209
Tune Position Error Integrator 209
Tune Torque Offset 209
Tune Velocity Error Integrator 209
Tune Velocity Feedforward 209

Tuning Direction Reverse 209
Tuning Speed 209

Tuning Torque 209

Tuning Travel Limit 209

Velocity Servo Bandwidth 209

Configuration Attributes
Axis Type 209

Motion Conversion Configuration

Conversion Constant 209
Motion Dynamics Configuration

Maximum Acceleration 209
Maximum Deceleration 209
Maximum Speed 209
Programmed Stop Mode 209
Fast Disable 209
Fast Shutdown 209
Fast Stop 209
Hard Disable 209
Hard Shutdown 209
Motion Homing Configuration

Active Homing
Active Immediate Home 120
Home Configuration Bits 209
Home Switch Normally Closed 209
Home Mode 209
Home Offset 209
Home Position 209
Home Return Speed 209
Home Sequence and Home Direction 209
Home Speed 209
Passive Homing
Passive Home with Marker 126
Passive Home with Switch 126
Passive Home with Switch then Marker 126
Passive Immediate Home 126
Motion Planner Configuration Attributes

Master Input Configuration Bits 209
Master Delay Compensation 209
Master Position Filter 209
Master Position Filter Bandwidth 209
Output Cam Execution Targets 209
Motion Unit Configuration Attributes

Average Velocity Timebase 209
Position Units 209
Position Unwind 209

Rotary Axis 209

Interface Attributes
Axis Configuration State 209

Axis Data Type 209

Consumed 209
Feedback 209
Generic 209
Servo 209
Servo Drive 209
Virtual 209
Axis Instance 209

Axis State 209

C2C Connection Instance 209

C2C Map Instance 209

Group Instance 209

Home Event Task Instance 209

Map Instance 209

Memory Usage 209

Memory Use 209

Module Channel 209

Module Class Code 209

Registration 1 Event Task Instance 209

Registration 2 Event Task Instance 209

Watch Event Task Instance 209

Module Fault Bit Attribute 209
Motion Status Attributes
Actual Acceleration 207, 209

Actual Position 209

Actual Velocity 209

Average Velocity 209

Command Acceleration 209

Command Position 209

Command Velocity 209

Interpolated Actual Position 209

Interpolated Command Position 209

Interpolation Time 209

Master Offset 209

Motion Status Bits 209

Registration Position 209

Registration Time 209

Start Master Offset 209

Start Position 209

Strobe Master Offset 209

Strobe Position 209

Watch Position 209

Servo Configuration Attributes

Absolute Feedback Enable 209

Absolute Feedback Offset 209

Axis Info Select 209

External Drive Type 209

Fault Configuration Bits 209

- Drive Fault Checking 209
- Drive Fault Normally Closed 209
- Hard Overtravel Checking 209
- Soft Overtravel Checking 209

LDT Calibration Constant 209

LDT Calibration Constant Units 209

LDT Length 209

LDT Length Units 209

LDT Recirculations 209

LDT Scaling 209

LDT Scaling Units 209

LDT Type 209

Servo Feedback Type 209

- A Quadrature B Encoder Interface 209
- Linear Displacement Transducer 209
- Synchronous Serial Interface 209

Servo Loop Configuration 209

Servo Polarity Bits 209

- Feedback Polarity Negative 209
- Servo Polarity Negative 209

SSI Clock Frequency 209

SSI Code Type 209

SSI Data Length 209

Servo Drive Attributes

Analog Input 209

Attribute Error Code 209

Attribute Error ID 209

Axis Control Bit Attributes 209

- Abort Process 209
- Change Cmd Reference 209
- Shutdown Request 209

Axis Info Select 209

Axis Response Bit Attributes 209

- Abort Event Acknowledge 209
- Abort Home Acknowledge 209
- Abort Process Acknowledge 209
- Change Pos Reference 209
- Shutdown Request Acknowledge 209

Commissioning Configuration Attributes

- Motor Inertia & Load Inertia Ratio 209

Commissioning Status Attributes
Test Direction Forward 209
Test Status 209
Tune Acceleration 209
Tune Acceleration Time 209
Tune Deceleration 209
Tune Deceleration Time 209
Tune Inertia 209
Tune Status 209
Drive Fault Bit Attributes 209
Drive Gains 340
Advanced Drive Gain Attributes 209
Output Notch Filter Frequency 209
Velocity Proportional Gain
Maximum Bandwidth 209
Drive Limits
Advanced Drive Limits 209
Continuous Torque Limit 209
Torque Limit 209
Drive Offsets
Backlash Reversal Error 209
Backlash Stabilization Window 209
Drive Fault Actions 209
Advanced Stop Action Attributes 209
Brake Engage Delay 209
Brake Release Delay 209
Resistive Brake Contact Delay 209
Drive Power Attributes
Bus Regulator ID 209
Power Supply ID 209
PWM Frequency Select 209
Drive Warning Bit Attributes 209
Cooling Error Warning 209
Drive Overtemperature Warning 209
Motor Overtemperature Warning 209
Overload Warning 209
Module Fault Bit Attributes 209
Module Hardware Fault 209
Timer Event Fault 209
Motor and Feedback Configuration
Aux Feedback Ratio 209
Feedback Configuration 209
Feedback Polarity 209
Feedback Interpolation 209
Feedback Resolution 209
Feedback Type 209
Feedback Units 209
Motor Data 209
Motor ID 209
SERCOS Error Code 209
Servo Drive Configuration Attributes
Advanced Scaling Attributes 209
Data Reference 209
Linear Scaling Unit 209
Scaling Type 209
Scaling Unit 209
Advanced Servo Configuration Attributes 209
Drive ID 209
Drive Polarity 209
Advanced Polarity Attributes 209
Custom Polarity 209
Negative Polarity 209
Positive Polarity 209
Drive Resolution 209
Drive Travel Range Limit 209
Drive Units 209
Fault Configuration Bits 209
Drive Enable Input Checking 209
Drive Enable Input Fault Handling 209
Hard Overtravel Checking 209
Soft Overtravel Checkin 209
Fractional Unwind 209
Linear Ball-Screw WITHOUT Aux Feedback Device 209
Linear Ball-Screw/Ball-Screw Combination WITH Aux Feedback Device 209
Rotary Gear-Head WITH Aux Feedback Device 209
Rotary Gear-Head WITHOUT Aux Feedback Device 209
Servo Loop Configuration 209
Servo Loop Block Diagrams
Auxiliary Dual Command Servo 338
Auxiliary Position Servo 335
Dual Command Feedback Servo 339
Dual Position Servo 336
Motor Dual Command Servo 337
Motor Position Servo 334
Torque Servo 339
Velocity Servo 339
Servo Drive Status Attributes
Acceleration Command 209
Aux Position Feedback 209

Bus Regulator Capacity 209

DC Bus Voltage 209

Drive Capacity 209

Drive Status Bit Attributes 209

Marker Distance 209

Motor Capacity 209

Motor Electrical Degrees 209

Negative Dynamic Torque Limit 209

Position Command 209

Position Error 209

Position Feedback 209

Position Integrator Error 209

Positive Dynamic Torque Limit 209

Power Capacity 209

Torque Command 209

Torque Feedback 209

Torque Limit Source 209

Velocity Command 209

Velocity Error 209

Velocity Feedback 209

Velocity Integrator Error 209

Servo Fault Configuration

Servo Fault Actions 209

Servo Gains

Acceleration Feedforward Gain 207, 209

Bandwidth Method 209

Integrator Hold Enable 209

Loop Gain Method 209

Maximum Bandwidth 209

Position Differential Gain 209

Position Integral Gain 209

Position Proportional Gain 209

Velocity Feedforward Gain 209

Velocity Integral Gain 209

Velocity Proportional Gain 209

- Backlash Reversal Error 209
- Backlash Stabilization Window 209
- Directional Scaling Ratio 209
- Maximum Bandwidth 209
- Output LP Filter Bandwidth 209
- Torque Scaling 209

Velocity Scaling 209

Servo Limits

Direct Drive Ramp Rate 209

Friction Compensation 209

Friction Compensation Window 209

Maximum Negative Travel 209

Maximum Positive Travel 209

Output Limit 209

Output Offset 209

Position Error Tolerance 209

Position Lock Tolerance 209

Torque Offset 209

Velocity Offset 209

Servo Loop Block Diagrams

Position Servo with Torque Servo Drive 332

Position Servo with Velocity Servo Drive 333

Servo Status Attributes
Acceleration Command 209
Attribute Error Code 209
Attribute Error ID 209
Aux Position Feedback 209
Axis Response Bit Attributes
Zero DAC Request Acknowledge 209
Commissioning Status Attributes
Test Direction Forward 209
Test Status 209
Tune Acceleration 209
Tune Acceleration Time 209
Tune Deceleration 209
Tune Deceleration Time 209
Tune Inertia 209
Tune Rise Time 209
Tune Speed Scaling 209
Tune Status 209
Marker Distance 209
Position Command 209
Position Error 209
Position Feedback 209
Position Integrator Error 209
Servo Fault Bit Attributes 209
Servo Output Level 209
Servo Status Bit Attributes 209
Velocity Command 209
Velocity Error 209
Velocity Feedbac 209
Velocity Integrator Error 209
Status Attributes
Output Cam Lock Status 209
Output Cam Pending Status 209
Output Cam Status 209
Output Cam Transition Status 209
Motion Axis Fault Reset 78, 101
Motion Axis Gear 78, 101
Motion Axis Home 78, 101
Motion Axis Jog 78, 101
Motion Axis Move 78, 101
Motion Axis Position Cam 78, 101
Motion Axis Shutdown 78, 101
Motion Axis Shutdown Reset 78, 101
Motion Axis Stop 78, 101
Motion Axis Time Cam 78, 101
Motion Calculate Cam Profile 78, 101
Motion Calculate Slave Values 78, 101
Motion Calculate Transform Position 78
Motion Change Dynamics 78, 101
motion control
add axis 40, 42
coordinate system 42, 62
program 42, 62
set up an axis 42
status information 42, 62
Motion Coordinated Change Dynamics 78, 101
Motion Coordinated Circular Move 78, 101
Motion Coordinated Linear Move 78, 101
Motion Coordinated Shutdown 78, 101
Motion Coordinated Shutdown Reset 78, 101
Motion Coordinated Stop 78, 101
Motion Coordinated Transform 78
Motion Direct Command Execution Error 85
Motion Direct Command Verification 83
Motion Direct Commands
Error Process 83
Transition States 85
Motion Direct Drive Off 78, 101
Motion Direct Drive On 78, 101
Motion Disarm Output Cam 78, 101
Motion Disarm Registration 78, 101
Motion Disarm Watch Position 78, 101
Motion Group Shutdown 78, 101
Motion Group Shutdown Reset 78, 101
Motion Group Stop 78, 101
Motion Group Strobe Position 78, 101
motion instructions 76

Coordinated Motion Instructions
Motion Coordinated Change Dynamics (MCCD) 78, 101
Motion Coordinated Circular Move (MCCM) 78, 101
Motion Coordinated Linear Move (MCLM) 78, 101
Motion Coordinated Shutdown (MCSD) 78, 101
Motion Coordinated Shutdown Reset (MCSR) 78, 101
Motion Coordinated Stop (MCS) 78, 101

Motion Configuration Instructions
Motion Apply Axis Tuning (MAAT) 78, 101
Motion Apply Hookup Diagnostic (MAHD) 78, 101
Motion Run Axis Tuning (MRAT) 78, 101
Motion Run Hookup Diagnostic (MRHD) 78, 101

Motion Event Instructions
Motion Arm Output Cam (MAOC) 78, 101
Motion Arm Registration (MAR) 78, 101
Motion Arm Watch Position (MAW) 78, 101
Motion Disarm Output Cam (MDOC) 78, 101
Motion Disarm Registration (MDR) 78, 101
Motion Disarm Watch Position (MDW) 78, 101

Motion Group Instructions
Motion Group Shutdown (MGSD) 78, 101
Motion Group Shutdown Reset (MGSR) 78, 101
Motion Group Stop (MGS) 78, 101
Motion Group Strobe Position (MGSP) 78, 101

Motion Move Instructions
Motion Axis Gear (MAG) 78, 101
Motion Axis Home (MAH) 78, 101
Motion Axis Jog (MAJ) 78, 101
Motion Axis Move (MAM) 78, 101
Motion Axis Position Cam (MAPC) 78, 101
Motion Axis Stop (MAS) 78, 101
Motion Axis Time Cam (MATC) 78, 101
Motion Calculate Cam Profile (MCCP) 78, 101
Motion Calculate Slave Values 78, 101
Motion Change Dynamics (MCD) 78, 101
Motion Redefine Position (MRP) 78, 101

Motion State Instructions
Motion Axis Fault Reset (MAFR) 78, 101
Motion Axis Shutdown (MASD) 78, 101
Motion Axis Shutdown Reset (MASR) 78, 101
Motion Direct Drive Off (MDF) 78, 101
Motion Direct Drive On (MDO) 78, 101
Motion Servo Off (MSF) 78, 101
Motion Servo On (MSO) 78, 101

overview 42, 62
Motion Move Instructions
Motion Axis Home (MAH)
Absolute Homing 119
Active Homing 120
Passive Homing 126

Motion Axis Jog (MAJ)
Velocity Profile Effects 90

Motion Redefine Position 78, 101
Motion Run Axis Tuning 78, 101
Motion Run Hookup Diagnostic 78, 101
Motion Servo Off 78, 101
Motion Servo On 78, 101
MOTION_GROUP data type 132
MOTION_GROUP structure 132

O

OK contact wire 330

S

SERCOS Drives
- **set up 42**

SERCOS interface drive
- **add to controller 33**

Specifications
- **1756-HYD02 Motion Module 14**
- **1756-M02AE Motion Module 14**
- **1756-M02AS Motion Module 14**
- **1756-M03SE, 1756-M08SE, & 1756-M16SE Motion Module 14**

W

Wiring connections 325
- **Connecting LDTs to the 1756-HYD02 module 325, 326**
- **Example diagram of 1756-HYD02 wiring 326**
- **home limit switch input 329**
- **OK contacts 330**

wiring diagrams 315

Rockwell Automation support

Rockwell Automation provides technical information on the web to assist you in using its products. At http://www.rockwellautomation.com/support you can find technical and application notes, sample code, and links to software service packs. You can also visit our Support Center at https://rockwellautomation.custhelp.com for software updates, support chats and forums, technical information, FAQs, and to sign up for product notification updates.

In addition, we offer multiple support programs for installation, configuration, and troubleshooting. For more information, contact your local distributor or Rockwell Automation representative, or visit http://www.rockwellautomation.com/services/online-phone.

Installation assistance

If you experience a problem within the first 24 hours of installation, review the information that is contained in this manual. You can contact Customer Support for initial help in getting your product up and running.

United States or Canada	1.440.646.3434
Outside United States or Canada	Use the Worldwide Locator available at http://www.rockwellautomation.com/locations, or contact your local Rockwell Automation representative.

New product satisfaction return

Rockwell Automation tests all of its products to ensure that they are fully operational when shipped from the manufacturing facility. However, if your product is not functioning and needs to be returned, follow these procedures.

United States	Contact your distributor. You must provide a Customer Support case number (call the phone number above to obtain one) to your distributor to complete the return process.
Outside United States	Please contact your local Rockwell Automation representative for the return procedure.

Documentation feedback

Your comments will help us serve your documentation needs better. If you have any suggestions on how to improve this document, complete the feedback form, publication RA-DU002.

Rockwell Otomasyon Ticaret A.Ş., Kar Plaza İş Merkezi E Blok Kat:6 34752 İçerenköy, İstanbul, Tel: +90 (216) 5698400

www.rockwellautomation.com

Power, Control and Information Solutions Headquarters

Americas: Rockwell Automation, 1201 South Second Street, Milwaukee, WI 53204-2496 USA, Tel: (1) 414.382.2000, Fax: (1) 414.382.4444
Europe/Middle East/Africa: Rockwell Automation NV, Pegasus Park, De Kleetlaan 12a, 1831 Diegem, Belgium, Tel: (32) 2 663 0600, Fax: (32) 2 663 0640
Asia Pacific: Rockwell Automation, Level 14, Core F, Cyberport 3, 100 Cyberport Road, Hong Kong, Tel: (852) 2887 4788, Fax: (852) 2508 1846

Rockwell Automation Publication MOTION-UM001D-EN-P - November 2015

Supersedes Publication MOTION-UM001C-EN-P - October 2014

Copyright © 2015 Rockwell Automation, Inc. All rights reserved. Printed in the U.S.A.